

UNCLASSIFIED

AD 295 693

*Reproduced
by the*

**ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA**



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

295 693

63-2-3

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

21G - 8

SICOSAS, AN IBM 7090 COMPUTER ASSEMBLY PROGRAM
FOR THE CDC 160-A COMPUTER

J. D. Drinan

17 January 1963

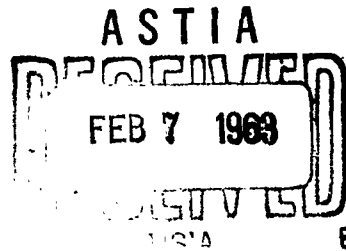
The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology; this work was supported by the U. S. Air Force under Air Force Contract AF 19(628)-500.

LEXINGTON

MASSACHUSETTS

TABLE OF CONTENTS

	Page
ABSTRACT	iii
INTRODUCTION	1
GENERAL DESCRIPTION	2
SICOM ASSEMBLY PROGRAM (SICAP)	4
USE OF \$ IN SICAP	5
SICAP LIBRARY SUBROUTINES	6
SICAP INFORMATION LAYOUT	7
MACHINE LANGUAGE ASSEMBLY PROGRAM (MLAP)	8
SICOSAS SPECIAL PURPOSE CARDS	9
SICAP	10
MLAP (MLAP. . . . OCTAL)	10
*(ASTERISK IN COLUMN 1)	10
NAME	10
ORG Y	10
DEC	11
OCT	11
BSS Y	11
EQU Y ± a	12
MASK	12
SQR	13
LOG	13
EXP	13
SIN	13
ATAN	13
CALL Y ± a, k	13
RETURN	13
END Y	13
SICOSAS LIMITATIONS AND RESTRICTIONS	13
ERROR PRINTOUTS	14
SICOSAS LOADER FOR THE 160-A	14
SICOSAS OPERATION NOTES	16
7090 PROCEDURE	16
160-A PROCEDURE	17
APPENDIX A: SICAP MNEMONIC OPERATION CODES	18
APPENDIX B: SICAP MNEMONICS ALPHABETIZED	33
APPENDIX C: MLAP MNEMONICS ALPHABETIZED	37
APPENDIX D: SAMPLE SICOSAS PROGRAM (SICAP ONLY)	43
APPENDIX E: SAMPLE SICOSAS PROGRAM (SICAP AND MLAP)	44
APPENDIX F: SICOSAS PROGRAM LISTING	45



AFESD - TDR - 63 - 17

ABSTRACT

SICOSAS is a symbolic assembly program designed to operate on the IBM 7090 for the purpose of merging many of the desirable features of two of CDC's programming systems, SICOM and OSAS-A, into a single programming system in order to make the CDC 160-A computer a more versatile tool for many applications. This memo is not intended as a self-sufficient primer. It is aimed at those who have had some background in the IBM 7090 computer and its associated SOS programming system as well as in the CDC 160-A computer with its OSAS-A and SICOM programming systems. Input cards are punched in the familiar SOS format. Output is an SOS-like program listing plus a binary copy of the assembled program on magnetic tape which is readily loaded and executed on the 160-A.

SICOSAS, AN IBM 7090 COMPUTER ASSEMBLY PROGRAM FOR THE CDC 160-A COMPUTER

INTRODUCTION

The present Lincoln Laboratory CDC 160-A computer configuration offers the user a relatively fast (6.4μ cycle time; 12.8μ add time) machine with a 16K, 12-bit-word magnetic core memory. Buffering is available as well as indirect addressing and program interrupt. Input-output media include paper tape, magnetic tape (IBM compatible), and a typewriter. Additionally, real time data may be processed using existing input-output facilities. A scope is also connected for display purposes.

To those who are using or have used some of the Laboratory's larger machines, there may be some disappointing aspect to the 160-A facility among the following.

- a. There are no machine-language multiply or divide instructions as such. These functions must be performed in other ways.
- b. There are no machine index registers.
- c. There is no floating point hardware.
- d. The basic machine-language compiler OSAS-A*, while symbolic and mnemonic, is quite separate from the interpretive programming system with which it may interplay.
- e. One of the two interpretive programming systems, SICOM[†], is decimal but non-mnemonic, non-symbolic, requires paper tape input for compilation, and affords no program listing.
- f. The versions of FORTRAN now available for the 160-A do not provide machine-language facility. [New versions of FORTRAN for the 160-A are under development[‡], but even these new versions have certain inadequacies. The availability of an adequate version of FORTRAN (e. g. , similar to 7090 FORTRAN) might well have obviated the current immediate need for SICOSAS.]

*Control Data Corporation Publication No. 507, "OSAS-A, The 160-A Assembly System."

† Scientific Computers, Inc. , Minneapolis, Minnesota, "SICOM, Scientific and Commercial Programming System for the Control Data 160-A Computer", June 1962.

‡ Control Data Corporation Publication No. 505, "160-A FORTRAN/General Information Manual."

SICOM has been described in the CDC literature* as having the following characteristics.

"SICOM for the Control Data 160-A computer is a general-purpose interpretive system utilizing floating point arithmetic. It effectively converts the 160-A from a binary, fixed-point machine with a 12-bit word length, into a decimal, floating-point machine with 10-decimal digit, plus exponent, word length. SICOM provides full arithmetic, indexing, and logical capabilities... as well as additional features which provide functions and SICOM or machine-language subroutines."

Certainly the prospect of going from a machine with no multiply/divide capability and having no index registers to the same machine possessing full arithmetic, indexing, logical and machine-language-dropout-and-return capabilities even at the reduced speed of an interpretive programming scheme is an enjoyable one for most applications.

It has been the primary intent then of the SICOSAS system to retain all of the desirable operational characteristics of SICOM while making programming for SICOM less of a chore. This has been done in part by by-passing the SICOM compiler completely in assembling the SICOM part of a SICOSAS program. A secondary feature of the SICOSAS system has been the elimination of a separate compilation of the machine-language portion of the SICOM program which would normally be done by OSAS-A since this compilation is also done by SICOSAS at the same time as the compilation of the main SICOM program.

SICOSAS then is really a dual assembler, each mnemonic and symbolic, each capable of independent operation. Each assembler and its ground rules will be considered separately.

GENERAL DESCRIPTION

Although the two assemblers in SICOSAS, SICAP (SICOM Assembly Program) and MLAP (Machine Language Assembly Program) may each be invoked to the exclusion of the other, the SICOSAS programming system was designed primarily for those who require interplay between the SICOM interpretive system with its large decimal capacity (10 decimal digits) and machine-language, hand-coded routines which perform special functions that cannot be accomplished in the SICOM language.

*Control Data Corporation Publication No. 502, "Programming Systems."

Before SICOSAS, a person with such a problem would prepare the SICOM code by writing it in an absolute numeric format for punching by the flexowriter. His only listing of the program would be the flexowriter printout of the absolute code which would simply be a probably neater copy of the original manuscript. No commentary of any kind would be retained beyond the original manuscript. Thus, a portion of the document from which he would work would appear something as follows:

```
1221003
0251004
0240720
:
036Y000
4
```

Next he would prepare his machine-language subroutines, coding these symbolically and with machine code mnemonics in the OSAS-A system. His origins for these subroutines would be equated to the effective machine location transferred to by the SICOM "transfer to machine language" instructions. Any reference to the main SICOM program would be made in absolute or quasi absolute in these subroutines since two separate unrelated compilers would prepare these codes for machine loading. Finally, the subroutines would be compiled by OSAS-A (via paper tape or card images on magnetic tape) and a final paper tape containing these subroutines would evolve for machine loading.

Having loaded the subroutines, the next step would be to load the SICOM compiler/interpreter and proceed to compile the main SICOM program from a flexo tape in the format described above over the previously-entered, hand-coded subroutines. If the SICOM program needed library mathematical subroutines, these would finally be loaded over everything else in the manner prescribed in the SICOM manual.

To accomplish the same result in the SICOSAS system, one codes the entire problem together as a single program. Programming is virtually entirely symbolic. Machine-language mnemonics have been carried over entirely from OSAS-A for constructing hand-coded subroutines. MLAP additionally can be made either a decimal compiler or octal compiler and may be switched back and forth from card to card if need be. For SICOM commands, since no mnemonics existed, a full set has been constructed. All of the basic SICOM mathematical library subroutines are now an integral part of SICOSAS so that any or all may be incorporated into the final program automatically at compile time thereby obviating their separate loading at execute time.

In summary, the procedure to be followed to arrive at a SICOSAS program

operating on the 160-A is as follows:

- a. Code the entire problem on card-room-provided coding forms.
- b. Use the card room facility for punching, verifying, etc.
- c. Compile the program on the 7090 either from tape (prestored) or from cards.
- d. Print out the BCD listing tape and save the binary tape (map of 160-A bank 1 core storage).
- e. At the 160-A call in assembled program for execution from magnetic tape with special SICOSAS loader.

This memo does not pretend to contain all the information required for a person to write a SICOSAS program. It is intended primarily for those who have at least a nodding acquaintance with the referenced literature, rather as a supplement describing a different approach to the solution of a certain class of problems. A limited number of SICOM Manuals are available for reference for those who wish to investigate this approach further. A listing of the SICOSAS program is included as Appendix F.

SICOM ASSEMBLY PROGRAM (SICAP)

The normal numeric format required by the bypassed SICOM compiler is of the following seven-digit form:

KOPADDR(1)

where K = the index register $0 \leq K \leq 9$,

and OP = the Operation code,

and ADDR = the SICOM address.

Two departures from this format occur for those commands in which DR of (1) above is the operation code and for those in which K of (1) above is the operation code.

The basic SICAP symbolic card layout is as follows:

Cols. 1-6	Location Field containing all blanks or an alpha-numeric symbol to be associated with this card. Col. 1 must be non-blank if a symbol is used. At least one character in the symbol must be non-numeric, and none may be + or - .
Col. 7	Blank
Cols. 8-15	Operation Field containing a mnemonic operation code beginning in Col. 8.
Cols. 16 +	Variable Field starting in Col. 16 is composed

of the address field and the tag (or index) field. The address field is separated from the tag field by a comma (,). The address field may or may not include an additive subfield. If it does contain this subfield, it is separated from the pure address by a + or - sign. A blank (b) in variable field is a terminating character. Thus, typical cards appear.

1	7 8	1 6
X R A Y	C L A	A B L E - 1 0 , 7
S Y M B O L	M P Y	P , 2
P	S T O	B O X + 2
	D V P	P I
S P L I T	I D V P C	2 3 4 5

Thus, for card XRAY above, if the symbol ABLE had previously been equated to SICAP location 1000, say, SICAP would consider K = 7, OP = 22 and ADDR = 990. Hence the equivalent numeric code 7220990 would be decoded and inserted into the machine cells assigned to location XRAY.

Cols. b+1 to 72

Commentary. Any combination of Hollerith characters.

Appendix A lists all of the SICAP mnemonic operation codes along with a description of each. Appendix B is an alphabetic listing of all mnemonics recognized by SICAP including all of the entries in Appendix A plus the SICAP pseudo-operation codes.

Appendix D shows a SICOSAS program in which all coding is in the SICAP language. This program is the symbolic, mnemonic, and documented equivalent of the SICOM program appearing on page 42 of the SICOM Manual. The first line of the listing contains the author identification field (J. D. Drinan Nov. 28, 1962.). The second line identifies the various columns in the listing. The four columns headed by MLOC show the machine-language (octal) location of the SICAP information. SILOC, of course, is the SICAP location (decimal) corresponding to MLOC. The K, OP, ADDR columns depict the decoded seven-digit SICOM instruction. At the far right of the listing appears simply a tabulation of the cards in the symbolic deck. The remainder of the columns contains an image (Cols. 1-72) of each input card.

USE OF \$ IN SICAP

In the original SICOM the notation Y000 was used to reference the SICOM accumulator. In SICAP this symbol has been replaced by \$ (dollar sign), thus the SICAP

instruction at location ABLE

ABLE OFLTB \$

would cause the contents of the accumulator to be output (in floating point followed by a tab) on the previously-selected device.

SICAP LIBRARY SUBROUTINES

The original SICOM system is composed in general of an interpreter/compiler program plus a library of relocatable arithmetic subroutines. Normally the subroutines required by the object program are manually loaded from paper tape into the absolute locations to which the object program will transfer. Control is then manually transferred to the SICOM dynamic start point in the main program for execution.

In SICOSAS a required subroutine is automatically incorporated into the object program upon the encounter of one of the five appropriate SUBROUTINE cards to be described here. Subroutine cards have the following format:

Location Field	A location symbol or all blank
Operation Field	Subroutine Designator
Variable Field	None

The subroutine designator is one of the following five specifying the square root, logarithmic, exponential, sin-cos, and arc tan subroutines, respectively.

SQR
LOG
EXP
SIN
ATAN

The effect of meeting one of these cards is to cause **SICAP** to insert one of these five basic subroutines into the object program at the place in the program where the subroutine designator card was met. If a symbol is present in the location field of the SUBROUTINE card, it will be entered into the symbol table and equated to the value of the location counter at the time of meeting. In any case, SICAP inserts a symbol of its own at this location identical to the symbol of the subroutine designator; i. e., SQR, LOG, EXP, SIN or ATAN. Thus, to enter a given subroutine, one may transfer (TSR) either to the program symbol used to identify the start of the subroutine or in the absence of such a symbol to the subroutine designator itself.

The reader's attention is invited to the table on page 55 of the SICOM Manual for information about arguments, entry points, etc., pertaining to these subroutines.

SICAP INFORMATION LAYOUT

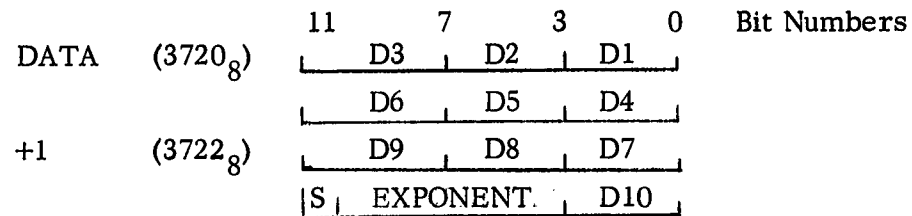
The reader is directed to page 83 of the SICOM Manual for information regarding the location and format of the SICOM pseudo accumulator.

The machine layout for SICAP commands and numeric data is given below since a) it does not appear in the SICOM Manual, and b) it is "must" information for those SICOSAS programs which employ SICAP for arithmetic computations and MLAP for logical manipulations of the results.

Suppose the following two cards appear in a SICAP program:

	ORG	1000
DATA	DEC	9876.543210

The SICAP digit numbering convention is $D_{10}D_9 \dots D_1$ where D_{10} of DATA is 9 and D_1 is 0. Each D_n occupies 4 bits weighted 8, 4, 2 and 1 from left to right. A SICAP datum is stored in two SICAP locations which take up four machine locations. As a result of the above cards, this numeric datum will be assembled into SICAP locations DATA and DATA+1 (these, of course, are absolute SICAP locations 1000_{10} and 1001_{10} which occupy machine locations 3720_8 through 3723_8) in the following format:



The octal representation of this number in cells 3720 through 3723 appears

$$\begin{aligned}
 1020 &= D_3D_2D_1 \\
 2503 &= D_6D_5D_4 \\
 4166 &= D_9D_8D_7 \\
 2111 &= S, \text{ EXP}, D_{10}
 \end{aligned}$$

The machine cell assigned to contain the first part ($D_3D_2D_1$) of the datum must be an even SICAP location. This requirement is automatically handled by SICAP and is of no concern to the user.

The SICAP command requires one SICAP location (two machine locations), thus the following two cards:

	ORG	100
START	CLA	START+1, 7

would cause information in the SICOM format KOPADDR to be assembled into the two equivalent machine locations 0310_8 and 0311_8 symbolically as follows:



In the above format bit 11 of the "ADDR" cell and bits 11 and 10 of the "KOP" cell are concerned with bank assignment and are of no interest to the programmer.

Numerically in SICAP these two cells appear

$0310 = 0101$
 $0311 = 0722$

When examined in core storage, the two cells appear

$0310 = 0145$
 $0311 = 0722$

MACHINE LANGUAGE ASSEMBLY PROGRAM (MLAP)

In SICOM exists an instruction K74ADDR (transfer to machine language) which, when executed by the SICOM interpreter program, results in uninhibited control being transferred to the SICOM location effective ADDR.

SICAP uses mnemonic CALL $Y \pm a, k$ for the same purpose. That is, the interpreter relinquishes control and the machine-language program beginning at SICAP effective location $Y \pm a, k$ is free to operate at machine-language speed.

MLAP was included in the SICOSAS system to bypass the need of a separate OSAS-A compilation of the machine-language subroutines needed in many SICOM programs. Every machine-language operation mnemonic used in OSAS-A is present in MLAP. Certain control pseudo operations are different and these will be discussed.

In using MLAP in a SICOSAS program one simply heads the portions of the program that are machine language with an MLAP card (see section entitled SICOSAS Special Purpose Cards) and codes in what is effectively the OSAS-A language. Return to the SICAP interpreter is accomplished through the use of a single machine-language macro instruction RETURN. (See SICOSAS Special Purpose Cards.)

The symbolic card format for MLAP is identical to that for SICAP, excepting, of course, that there is no tag (index) field in MLAP. The field layout then is

Cols. 1-6	Location Field
Col. 7	Blank
Cols. 8-15	Operation Field
Cols. 16 +	Variable Field

with all SICAP format ground rules for each field in effect.

Appendix E shows a SICOSAS program in which there is interplay between the SICAP and MLAP languages.

Because of the inherent differences in the requirements of OSAS-A, the basic assembly system for the 160-A computer and MLAP, whose function is to insert machine-language instructions in a larger over-all interpretive system, certain of the OSAS-A pseudo-operation functions have, in some cases, been dropped. For those pseudo operations which do have a parallel in MLAP, the reader is referred to the section entitled SICOSAS Special Purpose Cards. In summary, the status of the OSAS-A pseudo operations in MLAP is as follows:

<u>OSAS-A</u>	<u>MLAP</u>
ORG	ORG
PRG	None
CON	None
BLR	None
BSS	BSS
WAI	None
END	END
EQU	EQU
REM	*(Col. 1)
BNKX	None
SUPA	None
SUPB	None
BCD	None
BCDR	None
FLX	None
FLXR	None

Appendix C contains a complete alphabetized list of all mnemonics recognized by MLAP.

SICOSAS SPECIAL PURPOSE CARDS

The following cards are not of the command type listed in Appendix A. Some apply only when SICAP is controlling the assembly; others when MLAP is; still others when either is. All of these mnemonic, pseudo-operation codes occupy the operation field of the card unless stated otherwise.

SICAP

The SICAP card has no location field, no variable field. When a compilation is begun by SICOSAS, the assumption is that the initial cards conform to the SICAP mnemonic scheme. (See Appendix A.) When this mode is changed (by an MLAP card), the system is returned to the SICAP mode by a SICAP card. The SICAP card then specifies that the cards following it belong to the SICAP language, and hence, OSAS-A-type cards (recognized by MLAP) will be tagged as erroneous. A SICAP card read by SICAP has no effect.

MLAP (MLAP....OCTAL)

The MLAP card is a counterpart of the SICAP card. Upon meeting an MLAP card the mode of SICOSAS is changed to expect OSAS-A-type cards (see Appendix C) to follow until another SICAP card is met. The MLAP card has no location field. If the variable field (Cols. 16+) of the MLAP card is blank, the mode of MLAP is set to DECIMAL. If the variable field is alphabetic (such as OCTAL), the mode of MLAP is set to OCTAL. The variable fields of the cards following the MLAP (MLAP...OCTAL) card are converted decimally or octally depending on the mode established. An MLAP card read by MLAP has no effect.

*(ASTERISK IN COLUMN 1)

A card so punched has no effect on SICOSAS other than to reproduce the entire card on the output listing. This then is a remarks card and may be used either with SICAP or MLAP in control. Such cards may appear anywhere in the symbolic deck.

NAME

The NAME card has no location field. The purpose of the card is to identify the output listing. This is done by inserting the contents of columns 16 through 51 of the NAME card into an appropriate place in the first line of every page of the output listing. Normally the program author would use his name and the date. A NAME card may be read by either SICAP or MLAP. A later NAME card will supplant one read earlier.

ORG Y

An origin (ORG) card has the effect of resetting the location counter to the value of the decimal integer Y in the variable field. It is to be remembered that one basic SICOM instruction requires two machine cells. For this reason, Y is always considered a SICOM location so that the location counter $L_8 = 2Y_{10}$. Thus,

ORG 1000

resets L to 3720₈.

Any number of ORG cards may appear in the program. SICAP and MLAP treat this card identically.

DEC

The pseudo operation DEC applies only to SICAP. It is used to introduce floating point decimal data into the SICAP portion of the SICOSAS program. The location and variable fields of the DEC card conform to the general rules for these fields. The variable field contains a signed mantissa (+ may be omitted), a mandatory decimal point (·), and an optional signed exponent. Thus the following cards are valid examples of the DEC pseudo operation.

<u>Card</u>	<u>Will Convert To</u>
DEC 123.456E05	+ . 123456 x 10 ⁸
DEC -1.23456E-1	- . 123456 x 10 ⁰

It is to be remembered that in the variable field a blank is a terminating character so that

DEC 123.456Eb5 produces + . 123456 x 10³

SICOSAS automatically stores DEC data according to the SICOM ground rules for this type of input.

OCT

The pseudo operation OCT applies only to MLAP. This card is used when the mode of MLAP has been established as DECIMAL (see MLAP card), but it is desired to introduce an octal integer into the program. The variable field has four columns beginning in Col. 16. This field must contain an octal integer I, $0 \leq I \leq 7777$.

BSS Y

The "block starting with symbol" (BSS) pseudo operation is used to reserve a block of one or more words of core storage within the SICOSAS program. The variable field of the BSS card must be absolute. If SICAP is in control when the BSS is met, Y SICOM locations (2 x machine cells) are reserved. If MLAP meets the BSS, Y₁₀ or Y₈ machine locations are set aside depending on the mode of MLAP.

EQU Y ± a

The general rules for the location and variable field given previously apply to the pseudo operation EQU. Thus Y may be symbolic or absolute and a, if used, may be either octal or decimal. The EQU card, however, must appear in the symbolic deck ahead of any card using the symbol in the location field of the EQU card. Thus the following arrangement will compile correctly:

A	EQU	B
	CLA	A
	⋮	
B	DEC	0.

MASK

The MASK pseudo operation applies only to SICAP and is used in conjunction with the EXTR (K27 ADDR) instruction. The general rule for the location field applies. The variable field can be thought of as a floating point mask with up to ten sexadecimal numbers and an exponent completely analogous to the one in the DEC pseudo operation.

The sexadecimal numbers 0 through 9 correspond to the four-bit binary configurations 0000 through 1001.

U is 1010	X is 1101
V is 1011	Y is 1110
W is 1100	Z is 1111

Thus if a decimal datum is stored with the DEC pseudo operation

	⋮	
A	DEC	123.5
	⋮	
	CLA	A
B	EXTR	MASKA
	⋮	
MASKA	MASK	Z.ZZE1

The result of the extraction at location B would result in the number .235 appearing in the accumulator with E = 02; i. e. , 23.5 normalized.

Notice that the use of E, (·), and the blank are carried over from the DEC card.

SQR	
LOG	
EXP	These are SICAP Subroutine Designator cards explained under
SIN	SICAP Library Subroutines.
<u>ATAN</u>	

CALL Y ± a, k

This is a SICAP command (explained elsewhere but mentioned here in context with the RETURN macro) which links the SICAP portion of the SICOSAS program with the machine-language portion (MLAP). Control is transferred to the machine-language subroutine starting at effective SICAP location $Y \pm a, k$.

RETURN

RETURN is a macro-type, machine-language instruction that is the counterpart of the CALL command. RETURN may have a location field but has no variable field. When MLAP encounters a RETURN, it inserts the following three machine-language instructions into the program.

0040	SDC	Set Direct Bank
2006	LDD	Return Location
0030	IRJ	Jump to SICOM Interpreter

END Y

The variable field of the END pseudo operation is a symbolic location in the SICOSAS program to which control will first be passed upon loading the SICOSAS program from tape on the 160-A. The END card must be present; however, if Y is all blank, the dynamic start point will be set to the first symbolic location encountered in the program.

SICOSAS LIMITATIONS AND RESTRICTIONS

- a. The present version of SICOSAS was written with a two-bank (0 and 1) 160-A computer in mind. The SICOM compiler/interpreter program performs all of bank 0. The SICOSAS program is therefore limited to bank 1. This amount of storage is roughly equivalent to 1999 SICAP locations which must be shared with the machine-language portions of the program. SICOM, and therefore, SICAP, is enlargeable but at this writing this size limitation exists.
- b. Neither SICAP nor MLAP has the facility directly to accept alphabetic data (which would normally be used for headings by the final 160-A program).

In the case of SICAP, the user may enter this type of data through an input device, while with MLAP the octal equivalents may be compiled directly.

- c. Due to the manner in which the SICAP loader operates (see SICOSAS Loader), SICAP locations 0000 to 0005 (octal machine locations 0000 through 0013) may not be used directly by the SICOSAS program at compile time. This is to say that no program instruction or constant should be assigned to these cells; however, these locations may be freely referenced in the coding of the SICOSAS program through the use of the EQU card, for example, (see SICOSAS Special Purpose Cards) and freely used by the program in operation. While all other cells in bank one not explicitly containing the SICOSAS program will contain zeros, these first six SICAP locations will not.

ERROR PRINTOUTS

SICOSAS prints out error indications directly on the program listing as errors are encountered and continues with the assembly. Sources of error and the action taken are the following:

- a. Undefined symbol: The symbol is equated to zero.
- b. Multiply defined symbol: The symbol is assigned the value given it upon meeting it for the first time.
- c. Illegal operation code: A NOP replaces the illegal code.
- d. Out of range: In the 160-A machine instruction format EEXX, the XX which is evaluated as out of range is set to zero.
- e. Illegal additive field: The additive field is set to zero.
- f. OCT card variable field: Variable field is set to zero.
- g. Decimal data: Zero is stored for the illegal datum.

SICOSAS LOADER FOR THE 160-A

As a result of the assembly process on the 7090 computer, there exists a binary tape which when read into the 160-A computer will result in bank 1 containing the information necessary to operate the SICOSAS program.

To facilitate the loading of this tape into the 160-A, SICOSAS assembles the following SICAP instructions into the 7090 map of the 160-A bank 1 core memory in

such a fashion that upon read-in they will occupy SICAP locations 0000 through 0005 as follows:

0000	SELBIN	0	[Select binary tape 0]
1	ITAPE	0, 1	[Read in under control of index #1]
2	TSM4	4	[Error exit; transfer to 0004]
3	TSM4	[START]	[Read-in o. k. ; transfer to execute]
4	REW		[Try again]
5	TSM4	1	[Reread the tape]

When in the assembly process SICOSAS meets the END card, the absolute dynamic start point is determined and stored as the address of the TSM4 instruction in location 0003.

The problem with this tape at the 160-A is simply that of initiating its reading into location 0 of bank 1. This is accomplished by first reading in the SICOM interpreter into bank 0 and then have it call for paper tape input. (See Operation Notes.)

The paper tape which the SICOM interpreter reads contains the following equivalent SICAP instructions:

GRPIN	0	[Load following data starting in location 0]
SELBIN	0	[Select binary tape 0]
ITAPE	0, 1	[Read tape 0 under I. R. 1 control]
SIB	0, 1	[Set i base 0, I. R. 1]
SIL	1000, 1	[Set i limit 1999, I. R. 1]
ITYPE		[Select typewriter input]
OTYPE		[Select typewriter output]
TSM4	0	[Transfer to location 0000]
XEQ	2	[Start automatic computation at location 0002]

The instructions SELBIN through TSM4 0 are read by the SICOM interpreter/compiler and stored in locations 0000 through 0006 respectively. The XEQ 2 instruction transfers control to the SIB 0, 2 instruction at location 0002 under the guidance of the interpreter. The instructions SIL 1999, 1; ITYPE; and OTYPE are executed in turn whereupon control goes to location 0000 where the magnetic tape is selected and read-in to location 0000 and successive locations is initiated. The paper tape program is read over and the six order program at the beginning of the magnetic tape is in control. This program transfers to the SICOSAS program dynamic start point at the completion of a successful read-in, or, in the case of a faulty read-in, rewinds the magnetic tape and initiates another loading.

SICOSAS OPERATION NOTES

The procedure to be followed to have a SICOSAS program operate on the 160-A is a twofold one. First, one must run SICOSAS on the 7090 to obtain a) a printout of the listing of the assembled program, and b) a binary magnetic tape which contains a map of the assembled program as it will appear in bank 1 of the 160-A core memory. Second, one must load this binary magnetic tape into the 160-A for execution. These two phases will be discussed in the order outlined.

7090 PROCEDURE

SICOSAS (LA07 is the program identification number) is an SOS program which may be operated as such from the usual squeeze deck under control of "Modify and Load". For purposes of minimizing loading time and the number of mediary tapes required for loading, a self-sufficient binary version of SICOSAS has been recorded on magnetic tape. For use with this binary version of LA07 there is available a loader operating through the card reader which calls in SICOSAS from tape drive B9 (high density) and transfers to its dynamic start point for execution.

Irrespective of the method of loading, SICOSAS tape requirements (aside from those required by SOS if this method of loading is elected or the B9 if the binary version is used) are as follows:

A-7 (high density) BCD listing of assembled program.

A-8 (high density) BCD input tape (if program to be assembled has been prestored on tape).

B-7 (high density) Mediary tape.

B-8 (low density) Binary map of assembled program (input to 160-A).

When SICOSAS is first loaded it halts on an HPR at location 35540₈ to allow sense switch settings to be made as follows:

UP Input is on tape A-8.

SSW1

DOWN Input is on cards.

UP Print listing on-line (as well as off-line).

SSW3 (dynamic)

DOWN Do not print on-line.

UP When finished with this assembly, rewind the listing tape.

SSW6

DOWN Do not rewind the listing tape (stack output).

Upon the completion of an assembly, the binary map tape (B8) is rewound and must be replaced since no stacking is done on this tape. At this point the program

announces the completion of the job, halts and awaits the next run.

In addition to the initial HPR stop and the annotated "finished" stop, SICOSAS will stop on persistent input tape redundancies (HPR 1) and persistent output tape redundancies (HPR 2). The user is advised in either case to change tape and/or tape drive and start over. Pressing START at this point transfers to SICOSAS start point.

160-A PROCEDURE

The binary tape produced at the 7090 is mounted as tape 0 low density. The SICOM interpreter is now loaded (from SICOM master paper tape) into bank 0 as follows:

1. Clear and load to read in first block of paper tape. P = 0037, A = 7715, Z = 0.
2. Clear and run to read in remainder of interpreter program. P = 4562, A = 0005, Z = 7777.
3. SICOM is now ready to accept input via PETR. Place SICOSAS loader in reader, clear and run. The loader will now be loaded and will, in turn, load and execute the SICOSAS program.

JDD:pm

APPENDIX A

SICAP INSTRUCTIONS

The basic SICAP instruction card appears

1-----6	7	8-----15	16----->	b-----72
S Y M B O L		O P C O D E	Y ± a , k	C O M M E N T A R Y

In general the symbol (SYMBOL, above) in the location field (Cols. 1-6), if used, may contain one to six alphanumeric characters at least one of which must be non-numeric. The symbol must not contain + or - and must start in Column 1. The mnemonic operation code (OPCODE, above) in the operation field (Cols. 8-15) must start in Column 8. The variable field (starting in Col. 16 and terminating with a blank) has an address field (Y ± a, above) and may have a tag (or index) field which is separated from the address field by a comma (,). K, where used, must be a decimal integer $0 \leq K \leq 9$. The address field may contain an additive subfield (± a, above). The start of the additive field is signaled by the + or - sign. "a" must be a decimal integer $0 \leq a \leq 1999$. The address field (Y, above) may either be symbolic or a decimal integer $0 \leq Y \leq 9999$. If Y is absolute, no additive subfield is permitted.

The description of each instruction given below follows the ordering of instructions of the SICOM Command List on page (iv) of the SICOM manual. Attention is further invited to Appendix B which alphabetically lists all operations and pseudo operations recognized by SICAP along with references for each operation.

SIB	Y ± a, k	Set i Base	The base component of dimension i of index register k is set to the value Y ± a.
SID	Y ± a, k	Set i Difference	The difference component of dimension i of index register k is set to the value Y ± a.
SIL	Y ± a, k	Set i Limit	The limit component of dimension i of index register k is set to the value Y ± a.
DIB	Y ± a, k	Decrement i Base	This command operates on the i dimension of index register k. The base component is <u>decreased</u> by the amount of the difference component. The <u>resulting</u> base is then compared with the limit component. If the base is <u>equal to or larger than</u> the limit, Y ± a is then taken as the <u>location of the next command</u> to be executed. If the base is smaller, the command falls through, that is, the next command in

sequence is executed. During relative mode, operation is similar except $Y \pm a$ indicates the number of locations forward to the next command.

IIB	$Y \pm a, k$	Increment i Base	This command operates on the i dimension of index register k. The base component is <u>increased</u> by the amount of the difference component. The <u>resulting</u> base is then compared with the limit component. If the base is <u>smaller</u> or <u>equal</u> to the limit, $Y \pm a$ is then taken as the <u>location of the next command to be executed</u> . If the base is <u>larger</u> , the command falls through, that is, the next command in sequence is executed. During the relative mode, operation is similar except that $Y \pm a$ indicates the <u>number of locations forward to the next command</u> .
SJB	$Y \pm a, k$	Set j Base	The base component of dimension j of index register k is set to the value $Y \pm a$.
SJD	$Y \pm a, k$	Set j Difference	The difference component of dimension j of index register k is set to the value $Y \pm a$.
SJL	$Y \pm a, k$	Set j Limit	The limit component of dimension j of index register k is set to the value $Y \pm a$.
DJB	$Y \pm a, k$	Decrement j Base	This command operates similar to "Decrement i Base" except that the j dimension is involved.
IJB	$Y \pm a, k$	Increment j Base	This command operates similar to the command "Increment i Base" except that the j dimension is involved.
CADM	$Y \pm a, k$	Clear and Add Magnitude	The absolute value of contents of effective $Y \pm a$ is copied into the accumulator.
CLS	$Y \pm a, k$	Clear and Subtract	The contents of effective $Y \pm a$ are copied into the accumulator and the sign reversed.
CLA	$Y \pm a, k$	Clear and Add	The contents of effective $Y \pm a$ are copied into the accumulator.
IDVP	$Y \pm a, k$	Inverse Divide	The contents of effective $Y \pm a$ are divided by the contents of the accumulator.

DVP	$Y \pm a, k$	Divide	The contents of the accumulator are divided by the contents of effective $Y \pm a$.
MPY	$Y \pm a, k$	Multiply	The contents of the accumulator are multiplied by the contents of effective $Y \pm a$.
EXCH	$Y \pm a, k$	Exchange	The contents of the accumulator and the contents of effective $Y \pm a$ are interchanged.
EXTR	$Y \pm a, k$	Extract	The accumulator is extracted under control of the extractor in effective $Y \pm a$. All other accumulator bits are cleared. Each decimal digit is represented by four binary bits, therefore, any accumulator bit configuration may be detected by using sexadecimal numbers in the extractor. A "Z" digit in the extractor will extract the corresponding complete digit from the accumulator. The result of the extraction is in normalized (digitally) form with the proper exponent. To extract the sign, a negative extractor must be used. (See Appendix D, MASK pseudo operation.)
ADD	$Y \pm a, k$	Add	The contents of effective $Y \pm a$ are added to the contents of the accumulator.
SUB	$Y \pm a, k$	Subtract	The contents of effective $Y \pm a$ are subtracted from the contents of the accumulator.
ADM	$Y \pm a, k$	Add Magnitude	The absolute value of the contents of effective $Y \pm a$ are added to the contents of the accumulator.
ISUB	$Y \pm a, k$	Inverse Subtract	The contents of the accumulator are subtracted from the contents of the effective $Y \pm a$. The result is in the accumulator. Effective $Y \pm a$ is unchanged.
RAD	$Y \pm a, k$	Replace Add	The contents of the accumulator are added to the contents of effective $Y \pm a$ and the result is stored in effective $Y \pm a$.
CAS	$Y \pm a, k$	Compare Accumulator with Storage	The contents of the accumulator are compared with the contents of effective $Y \pm a$. (1) If effective $Y \pm a$ is the smaller of the two, control goes to the next command in

sequence. (2) If the two are equal, control goes to the second location down, i.e., if the command is at XRAY, an equal control will go to XRAY+2. (3) If effective $Y \pm a$ is larger, control goes to the third command down, i.e., XRAY+3. This command cannot be executed from the last three locations in a bank (i.e., 1997, 1998, and 1999) due to the multiple exits.

OFLTB	$Y \pm a, k$	Output Floating Point and Tab	The contents of effective $Y \pm a$ are put out as a floating point number on the previously selected device. The format is sign, period, 10 digits, exponent, followed by a tab.
OFLCR	$Y \pm a, k$	Output Floating Point and Carriage Return	The contents of effective $Y \pm a$ are put out as a floating point number on the previously selected device. The format is sign, period, 10 digits, exponent, followed by a carriage return.
OFXTB	$Y \pm a, k$	Output Fixed Point and Tab	The contents of effective $Y \pm a$ are put out as a fixed point number on the previously selected device. The format is sign, m integer digits, period, n fractional digits, followed by a tab. m and n are set by "OFORM" Command.
OFXCR	$Y \pm a, k$	Output Fixed Point and Carriage Return	The contents of effective $Y \pm a$ are put out as a fixed point number on the previously selected device. The format is sign, m integer digits, period, n fractional digits, followed by a carriage return. m and n are set by "OFORM" Command.
OFORM	$Y \pm a, k$	Set Output Format	If effective $Y \pm a, k$ is represented as a decimal integer ADDR, m, $0 \leq m \leq 18$ is set to AD digits and n, $0 \leq n \leq 18$ is set to DR digits. (See OFXTB and OFXCR) AD plus DR cannot exceed 18. If DR is zero, the period is not put out.
OCRTB	$Y \pm a, k$	Output Carriage Returns and Tabs	If effective $Y \pm a, k$ is represented as a decimal integer ADDR, AD carriage returns followed by DR tabs are put out.
OTBNO	$Y \pm a, k$	Output Tabulating Number	The value effective $Y \pm a$ is put out without a period followed by a tab. The largest number which may be put out is 9999. This command is useful for automatic numbering of lines and columns of output.
OCMND	$Y \pm a, k$	Output a Command from Memory	The command in location effective $Y \pm a$ is put out in the standard form (K OP $Y \pm a$) followed by a carriage return.

WRFILE $Y \pm a, k$

Write File Number on Magnetic Tape

An end of file mark and a record containing the number effective $Y \pm a$ is written on the previously-selected tape drive. Effective $Y \pm a$ is the file number identifying the file which follows. The normal exit is $L + 2$. The conditional exit ($L + 1$) is utilized in the event that end of tape has been sensed during this or a previous operation. Since the search operation assumes no files are opened after the end of tape spot, it is recommended that if the end of tape spot is sensed, the file number be erased and the tape ended (backspace 2 records and write the end of tape code number). If bad tape or a drive malfunction prevents the writing of a file number anywhere between the initial position of the tape and the end of the tape, the drive will "hang up" on the conductive trailer at the end of the tape. File numbers must be in increasing size as you proceed to the end of the tape. Zero should not be used as a file number.

SRFILE $Y \pm a, k$

Search Magnetic Tape for File Number

A search for file number effective $Y \pm a$ is initiated on the previously-selected tape drive. Upon the conclusion of a successful search, the next command executed is the second one in sequence ($L + 2$) (normal exit). The conditional exit ($L + 1$) is used if either of the following occur:

- (1) Ten attempts to read a file number have failed due to a parity error. The accumulator is set to zero prior to exiting in this case.
- (2) A reflective spot was sensed during the search operation (Load Point or End of Tape). The first file number after the spot is placed in the accumulator.

The search operation starts by backspacing to and reading the first file number back. A comparison of this "tape" file number with $Y \pm a$ determines the direction of the search.

If the "tape" number is larger than effective $Y \pm a$, the search will be in the direction of the load point.

If the "tape" number is smaller than $Y \pm a$ or if the load point is sensed, the search will be in the direction of the end of the tape. If a subsequent file number indicates that the direction should be changed again, the standard error indication of 0047 is given. Cycling the run switch after this indication will cause the search to restart.

Example: The search for 101 among consecutive numbers of 98, 100 and 103 would cause error 0047.

GRPIN	Y ± a, k	Group Input	Input is transferred to effective Y ± a and consecutive locations. (See SICOM Manual, paragraph II. B, 6, p. 14, "Normal Input".)
SNGIN	Y ± a, k	Single Input	Input is transferred only to location effective Y ± a. One word of any type of input may be entered. The input is terminated by either a carriage return or a tab.
IOCTAL	Y ± a, k	Input Octal Tape	Tape punched by "OOCTAL" Command is read into the receiving field starting at effective Y ± a. The field must fall entirely within a bank. The location of the receiving field may differ from the original source field.
OALPHA	Y ± a, k	Output Alphanumeric Data	The contents of address effective Y ± a and effective Y ± a + 1 are put out as 8 alphanumeric characters on the previously-selected device. This instruction is not affected by the relative mode.
LDANR	Y ± a, k	Load Alpha Numeric Register (A. N. R.)	The contents of effective Y ± a are loaded into A. N. R. as alphanumeric data. (See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
CMPANR	Y ± a, k	Compare Alpha Numeric Register (A. N. R.)	(See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
MRGANR	Y ± a, k	Merge into Alpha Numeric Register (A. N. R.)	(See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
EXTANR	Y ± a, k	Extract Alpha Numeric Register (A. N. R.)	(See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
TZE	Y ± a, k	Transfer on Accumulator Zero	If the contents of the accumulator is zero, a mark 60 jump to effective Y ± a occurs. Otherwise, the normal sequence is continued.
TNZ	Y ± a, k	Transfer on Accumulator Non-Zero	If the contents of the accumulator is not zero, a mark 61 jump to effective Y ± a occurs. Otherwise, normal sequence continues.
TPL	Y ± a, k	Transfer on Accumulator Plus	If the accumulator contains a positive quantity (including zero), a mark 62 jump to effective Y ± a occurs. Otherwise, normal sequence continues.

TMI	$Y \pm a, k$	Transfer on Accumulator Minus If the accumulator contains a negative quantity, a mark 63 jump to effective $Y \pm a$ occurs. Otherwise, normal sequence continues.
TSM4	$Y \pm a, k$	Transfer and Set Mark 4 The next command in sequence is marked for a "return to mark 4" return. Control goes to effective $Y \pm a$.
TSM5	$Y \pm a, k$	Transfer and Set Mark 5 The next command in sequence is marked for a "return to mark 5" return. Control goes to effective $Y \pm a$.
TSM6	$Y \pm a, k$	Transfer and Set Mark 6 The next command in sequence is marked for a "return to mark 6" return. Control goes to effective $Y \pm a$.
TNR7	$Y \pm a, k$	Transfer Non-Relative and Set Mark 7 The next command in sequence is marked for a "return to mark 7" return. Control goes to effective $Y \pm a$. This command is <u>not</u> affected by the relative mode. This makes a jump from relative subroutine to fixed area possible.
TSR	$Y \pm a, k$	Transfer to Subroutine This command operates the same as other unconditional transfer commands. It is also used to enter standard SICOM subroutines, thereby disturbing any 70 mark which may have been set previously.
TSJ1	$Y \pm a, k$	Transfer on Selective Jump Switch 1 On If selective jump switch number one is ON, the next consecutive command is marked for a return to selective jump one (RSJ1) return and control goes to the location effective $Y \pm a$. If the jump switch is not ON, the normal sequence is continued.
TSJ2	$Y \pm a, k$	Transfer on Selective Jump Switch 2 On This command operates similar to the one above except that jump switch two is used and the return is "return to selective jump two" (RSJ2).
TSJ3	$Y \pm a, k$	Transfer on Selective Jump Switch 1 and 2 Both On If both jump switch one and two are ON, the next consecutive command is marked for a return to "return to selective jump three (RSJ3)". Control goes to effective $Y \pm a$. If either jump switch is OFF, the next command in sequence is executed.
CALL	$Y \pm a, k$	Transfer to Machine Language Subroutine Control is transferred to the machine language subroutine which starts at effective location $Y \pm a$.

TBR	$Y \pm a, k$	Transfer Back Relative
		Control goes backwards $Y \pm a$ effective locations regardless of the addressing mode. No mark is set.
STOANR	$Y \pm a, k$	Store A. N. R.
		The contents of A. N. R. are copied into effective $Y \pm a$ and effective $Y \pm a + 1$ as alphanumeric data. A. N. R. is unchanged.
STO	$Y \pm a, k$	Store Accumulator
		The contents of the accumulator are copied into effective $Y \pm a$ and effective $Y \pm a + 1$. The accumulator remains unchanged.
NOP		No Operation
		This command may be used to reserve a location to be filled conditionally. No operation is performed. The computer proceeds to the next location.
STOP1		Selective Stop 1
		If selective stop switch 1 is on, computation stops with Panel display A = 0001. Resume computation by cycling run switch. If the switch is off, the command has no effect. Computation continues in normal sequence regardless of switch setting.
STOP2		Selective Stop 2
		Same as Selective Stop 1 except under control of Switch 2, Panel Display: A = 0002.
HALT3		Stop Display 3
		Computation stops with Panel Display: A = 0003. It may be resumed at the next sequential location by cycling the run switch.
HALT4		Stop Display 4
		Same as Stop Display 3 only A = 0004.
HPRIN		Halt and Await Input
		If Flexowriter input has been selected, computation stops with A = 0005 and input is <u>when</u> the run switch is cycled. If typewriter input has been selected, input is gated. (See SICOM Manual, paragraph IV.a, p. 71, "Operating Modes.")
REL		Select Relative Mode
		All subsequent commands are interpreted in the relative addressing mode. Termination is by execution of a "SELECT ABSOLUTE MODE" (ABS), a master clear and run or "Halt and Await Input".

ABS		Select Absolute Mode
		Computation returns to the normal absolute addressing mode.
FAST		Select Non-Trace Mode
		Tracing is discontinued. This command is normally used to avoid tracing through SICOM subroutines. During non-trace operation this command is the same as a "NO OPERATION". This also discontinues the step mode. Both operations are restored upon execution of "Reset Trace" or a master clear and run.
TRACE		Reset Trace Mode
		Restores the trace and step modes which were in effect prior to the execution of a "SELECT NON-TRACE" (FAST). This command is normally placed at the end of a SICOM subroutine.
ZEROIR		Clear Index Registers
		This command sets all components of all index registers to zero.
IFLEX		Select Flexowriter Input
		All subsequent normal input commands will be via Flexowriter tape.
OFLEX		Select Flexowriter Output
		All subsequent normal output will be punched on Flexowriter tape.
ITYPE		Select Typewriter Input
		All subsequent normal input commands will be via the typewriter.
OTYPE		Select Typewriter Output
		All subsequent normal output will be printed via the typewriter.
OPRINT		Select Printer
		(Not Available.)
SELBCD		Select Tape to Printer
		All subsequent normal output is put on magnetic tape drive 1 suitable for off line magnetic tape to printer listing. Only tape drive 1 and print channel 0 can be used. Each line to be printed is recorded as a 120 Binary Coded Decimal character record.
SELBIN	Y \pm a, k	Select Magnetic Tape Drive Y \pm a, k
		If k = 0, the j base of index k is added to Y \pm a. This new value of Y \pm a (or the original value if k = 0) is the number of the drive which will be used in all subsequent

tape operations. The accumulator is not disturbed. This command automatically selects odd parity. $Y \pm a$ may have values of 1, 2, 3, or 4. If the power is off on drive $Y \pm a$, or if no drive has been set to $Y \pm a$, the computer hangs up with the select light on. The execution of 0 00 0017 "Select Tape to Printer" SELBCD voids all prior magnetic tape selections.

BSR $Y \pm a$

Backspace $Y \pm a$ Records

If $Y \pm a \neq 0$, the previously-selected drive will be backspaced $Y \pm a$ records. Note that an end of file mark and file numbers are each considered a record, and that no indication of load point or end of tape is given. $Y \pm a$ may have values up to 99. Backspacing beyond the load point will cause the drive to "hang up" on the conductive leader.

REW

Rewind

The previously-selected drive will rewind to the load point. A rewind from the load point has no effect.

OTAPE 0, k

Write on Magnetic Tape

The field defined by index register k (i Base = 1st location, i Difference = length) is copied on the previously-selected tape drive as one record. The accumulator is not disturbed by either the normal ($L + 2$) or the conditional ($L + 1$) exit. The conditional exit indicates that the end of tape spot was sensed on this or a previous command.

If a parity error occurs during a write operation, the record is erased, 6 inches of tape skipped and the record is re-written. This process continues until the record is properly written or the conductive leader is reached. Due to the double exit, this command cannot be executed from the last two locations in a bank, i. e., 1998 and 1999.

ITAPE 0, k

Read Magnetic Tape

One record of information is read from the previously-selected tape drive. It is read into the field defined by index register K (i base = starting location, i difference = the length). If the tape record is longer than the field, the remainder of the record will be read but not stored in memory.

If the record is shorter than the field, all of the remainder of the field will be unchanged except the location immediately following the end of the record. This location will be altered. The normal exit ($L + 2$) does not disturb the accumulator. The conditional exit ($L + 1$) is utilized for the following two conditions:

(1) Ten attempts to read the record are unsuccessful due to a parity error. The accumulator is set to zero prior to exiting in this case. The record is stored in memory as it was read on the last attempt.

(2) The end of file was read. In this case, the next file number is placed into the accumulator ready for possible test to determine if the end of the tape has been reached. No end of tape indication as such is given.

An attempt to read beyond the last record on the tape will result in the drive "hanging up" on the conductive trailer. Due to the double exit, this command cannot be executed from the last two locations in a bank, i. e. , 1998 and 1999.

BLCPY 0, k

Block Copy

The source field defined by index register k is copied into a receiving field which starts at the address specified by the j base of index k. If the two fields overlap, the j base must be smaller than the i base. The two fields may be in different banks. (See SICOM Manual, paragraph II. B. 19, p. 30, "Block Operations. ")

BLCLR 0, k

Block Clear

The field defined by index register k is cleared to zero. (See SICOM Manual, paragraph II. B. 19, p. 30, "Block Operations. ")

OOCTAL 0, k

Output Octal Tape

The field defined by index register k is punched, in bi-octal form, followed by two blank frames and the check sum (two frames). This tape may be read with the IOCTAL command. A one-inch trailer is also punched.

OSPEC 0, k

Output Special Tape

The field defined by index k is punched in bi-octal form in the format shown on page 31 of the SICOM Manual.

ISPEC

Input Special Tape

Reads tape punched by the command OSPEC. Reading stops when a leader over 6" long is detected.

RZE

Return to TZE

Control is returned to the location +1 of the most recently executed transfer on accumulator zero (TZE) instruction.

RNZ

Return to TNZ

Control is returned to the location +1 of the most recently executed transfer on accumulator non-zero (TNZ) instruction.

RPL	Return to TPL Control is returned to the location +1 of the most recently executed transfer on accumulator positive (TPL) instruction.
RMI	Return to TMI Control is returned to the location +1 of the most recently executed transfer on accumulator negative (TMI) instruction.
RSM4	Return to TSM4 Control is returned to the location +1 of the most recently executed transfer and set Mark 4 (TSM4) instruction.
RSM5	Return to TSM5 Control is returned to the location +1 of the most recently executed transfer and set Mark 5 (TSM5) instruction.
RSM6	Return to TSM6 Control is returned to the location +1 of the most recently executed transfer and set Mark 6 (TSM6) instruction.
RNR7	Return to TNR7 Control is returned to the location +1 of the most recently executed transfer non-relative and set Mark 7 (TNR7) command.
RSR	Return to TSR Control is returned to the location +1 of the most recently executed transfer to subroutine (TRS) instruction.
RSJ1	Return to TSJ1 Control is returned to the location +1 of the most recently executed transfer if selective jump switch one is on (TSJ1) command.
RSJ2	Return to TSJ2 Control is returned to the location +1 of the most recently executed transfer if selective jump switch two is on (TSJ2) command.
RSJ3	Return to TSJ3 Control is returned to the location +1 of the most recently executed transfer if selective jump switch three is on (TSJ3) command.
OLOC	Output Last Location Location of last command executed is put out on the previously-selected device.

PUSTOP

Punch Stop, Check and Leader

The following are punched on tape:

- (1) A stop code,
- (2) The check sum of all normal output punched since the last "Halt Select Manual Mode" or since the last "Punch Stop, Check, and Leader."
- (3) A one-inch leader,

This is useful for Flexowriter output exclusively.

CLAC* Y ± a

Clear and Add Constant

The value Y ± a is loaded into the accumulator. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

CLSC* Y ± a

Clear and Subtract Constant

The value minus Y ± a is loaded into the accumulator. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

IDVPC* Y ± a

Divide into Constant

The contents of the accumulator are divided into the value Y ± a. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

DVPC* Y ± a

Divide by Constant

The accumulator is divided by the value Y ± a. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

MPYC* Y ± a

Multiply by Constant

The accumulator is multiplied by the value Y ± a. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

ADDC* Y ± a

Add Constant

The value Y ± a is added to the accumulator. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

SUBC* Y ± a

Subtract Constant

The value Y ± a is subtracted from the accumulator. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

*The variable field must not contain a decimal point (·).

ISUBC *	$Y \pm a$	Subtract from Constant	The accumulator is subtracted from the value $Y \pm a$. The $Y \pm a$ of these commands is <u>not</u> an address but instead a <u>numeric value</u> between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.
SHIFT*	$Y \pm a$	Shift Accumulator	If $Y \pm a$ is represented as a decimal integer ADDR, the accumulator is multiplied by 10 to the AD power (left shift) and divided by 10 to the DR power (right shift).
SETFWA	$Y \pm a$	Set First Address for Trace	An internal switch is set so that during the trace mode tracing starts after the execution of the command in location $Y \pm a$.
SETLWA	$Y \pm a$	Set Last Address for Trace	The address at which tracing will terminate is set to $Y \pm a$.
PXAIB	0, k	Place Index k (i Base) in Accumulator	The i base component of index register k replaces the contents of the accumulator as a positive floating point number.
PXAID	0, k	Place Index k (i Difference) in Accumulator	The i difference component of index register k replaces the contents of the accumulator as a positive floating point number.
PXAIL	0, k	Place Index k (i Limit) in Accumulator	The i limit component of index register k replaces the contents of the accumulator as a positive floating point number.
PXAJB	0, k	Place Index k (j Base) in Accumulator	The j base component of index register k replaces the contents of the accumulator as a positive floating point number.
PXAJD	0, k	Place Index k (j Difference) in Accumulator	The j difference component of index register k replaces the contents of the accumulator as a positive floating point number.
PXAJL	0, k	Place Index k (j Limit) in Accumulator	The j limit component of index register k replaces the contents of the accumulator as a positive floating point number.

*The variable field must not contain a decimal point (·).

PAXIB	0, k	Place the Accumulator in the i Base Component of Index Register k The integral portion of the accumulator is copied into the base component of index register k. The accumulator is unchanged.
PAXID	0, k	Place the Accumulator in the i Difference Component of Index Register k The integral portion of the accumulator is copied into the i difference component of index register k. The accumulator is unchanged.
PAXIL	0, k	Place the Accumulator in the i Limit Component of Index Register k The integral portion of the accumulator is copied into the i limit component of index register k. The accumulator is unchanged.
PAXJB	0, k	Place the Accumulator in the j Base Component of Index Register k The integral portion of the accumulator is copied into the j base component of index register k. The accumulator is unchanged.
PAXJD	0, k	Place the Accumulator in the j Difference Component of Index Register k The integral portion of the accumulator is copied into the j difference component of index register k. The accumulator is unchanged.
PAXJL	0, k	Place the Accumulator in the j Limit Component of Index Register k The integral portion of the accumulator is copied into the j limit component of index register k. The accumulator is unchanged.
XEQ	Y \pm a	Execute Automatic computation is begun starting at effective location Y \pm a under the guidance of the SICOM interpreter.

APPENDIX B

Below are listed alphabetically all of the mnemonics recognized by SICAP. The + notation for the pseudo operations refers the reader to the section entitled SICOSAS SPECIAL PURPOSE CARDS. Where applicable, the original seven-digit SICOM numeric format is included as well as a page reference to the SICOM manual. Finally, for each mnemonic operation, a page in Appendix A is referenced where a complete description is given.

<u>SICAP MNEMONIC</u>	<u>SICOM FORMAT</u>	<u>SICOM MANUAL PAGE #</u>	<u>SICOSAS APPENDIX A PAGE #</u>
ABS	0 00 0007	24	26
ADD	*K 30 ADDR	10	20
ADDC	6 01 ADDR	33	30
ADM	*K 32 ADDR	10	20
ATAN			+
BLCLR	K 00 0025	30	28
BLCPY	K 00 0024	30	28
BSR	0 00 AD19	27	27
BSS			+
CADM	*K 20 ADDR	09	19
CALL	*K 74 ADDR	22	24
CAS	*K 35 ADDR	10	20
CLA	*K 22 ADDR	09	19
CLAC	1 01 ADDR	33	30
CLS	*K 31 ADDR	10	19
CLSC	2 01 ADDR	33	30
CMPANR	*K 55 ADDR	18	23
DEC			+
DIB	*K 05 ADDR	07	18
DJB	*K 15 ADDR	08	19
DVP	*K 24 ADDR	09	20
DVPC	4 01 ADDR	33	30
END			+
EQU			+
EXCH	*K 26 ADDR	09	20
EXP			+
EXTANR	*K 57 ADDR	19	23
EXTR	*K 27 ADDR	09	20
FAST	0 00 0008	24	26
GRPIN	K 50 ADDR	15	23
HALT3	0 00 0003	23	25
HALT4	0 00 0004	23	25
HPRIN	0 00 0005	24	25
IDVP	*K 23 ADDR	09	19
IDVPC	3 01 ADDR	33	30
IFLEX	0 00 0011	25	26

· Affected by relative mode.

<u>SICAP MNEMONIC</u>	<u>SICOM FORMAT</u>	<u>SICOM MANUAL PAGE #</u>	<u>SICOSAS APPENDIX A PAGE #</u>
IIB	*K 06 ADDR	08	19
IJB	*K 16 ADDR	08	19
IOCTAL	K 52 ADDR	17	23
ISPEC	0 00 0029	31	28
ISUB	*K 33 ADDR	10	20
ISUBC	8 01 ADDR	34	31
ITAPE	K 00 0021	28	27
ITYPE	0 00 0013	25A	26
LDANR	*K 54 ADDR	18	23
LOG			+
MASK			+
MLAP			+
MPY	*K 25 ADDR	09	20
MPYC	5 01 ADDR	33	30
MORGANR	*K 56 ADDR	19	23
NAME			+
NOP	0 00 0000	23	25
OALPHA	K 53 ADDR	17	23
OCMND	K 45 ADDR	12	21
OCRTB	K 43 ADDR	12	21
OFLCR	K 37 ADDR	11	21
OFLEX	0 00 0012	25A	26
OFLTB	K 36 ADDR	11	21
OFORM	K 42 ADDR	12	21
OFXCR	K 41 ADDR	11	21
OFXTB	K 40 ADDR	11	21
OLOC	0 00 0075	32	29
OOCTAL	K 00 0027	30	28
OPRINT	0 00 0016	25A	26
ORG			+
OSPEC	K 00 0028	31	28
OTAPE	K 00 0020	27	27
OTBNO	K 44 ADDR	12	21
OTYPE	0 00 0014	25A	26
PAXIB	K 00 0223	29	32
PAXID	K 00 0323	29	32
PAXIL	K 00 0423	29	32
PAXJB	K 00 1223	29	32
PAXJD	K 00 1323	29	32
PAXJL	K 00 1423	29	32
PUSTOP	0 00 0079	32	30
PXAIB	K 00 0222	29	31
PXAID	K 00 0322	29	31
PXAIL	K 00 0422	29	31
PXAJB	K 00 1222	29	31
PXAJD	K 00 1322	29	31
PXAJL	K 00 1422	29	31
RAD	*K 34 ADDR	10	20
REL	0 00 0006	24	25

*Affected by relative mode.

<u>SICAP MNEMONIC</u>	<u>SICOM FORMAT</u>	<u>SICOM MANUAL PAGE #</u>	<u>SICOSAS APPENDIX A PAGE #</u>
REW	0 00 0019	27	27
RMI	0 00 0063	32	29
RNR7	0 00 0067	32	29
RNZ	0 00 0061	31	28
RPL	0 00 0062	31	29
RSJ1	0 00 0071	32	29
RSJ2	0 00 0072	32	29
RSJ3	0 00 0073	32	29
RSM4	0 00 0064	32	29
RSM5	0 00 0065	32	29
RSM6	0 00 0066	32	29
RSR	0 00 0070	32	29
RZE	0 00 0060	31	28
SELBCD	0 00 0017	26	26
SELBIN	K 00 0018	27	26
SETFWA	Y 01 ADDR	34	31
SETLWA	Z 01 ADDR	34	31
SHIFT	9 01 ADDR	34	31
SIB	K 02 ADDR	07	18
SICAP			+
SID	K 03 ADDR	07	18
SIL	K 04 ADDR	07	18
SIN			+
SJB	K 12 ADDR	08	19
SJD	K 13 ADDR	08	19
SJL	K 14 ADDR	08	19
SNGIN	K 51 ADDR	17	23
SQR			+
SRFILE	K 47 ADDR	13	22
STO	*K 77 ADDR	22	25
STOANR	*K 76 ADDR	22	25
STOP1	0 00 0001	23	25
STOP2	0 00 0002	23	25
SUB	*K 31 ADDR	10	20
SUBC	7 01 ADDR	33	30
TBR	K 75 ADDR	22	25
TMI	*K 63 ADDR	21	24
TNR7	K 67 ADDR	21	24
TNZ	*K 61 ADDR	20	23
TPL	*K 62 ADDR	21	23
TRACE	0 00 0009	24	26
TSJ1	*K 71 ADDR	21	24
TSJ2	*K 72 ADDR	22	24
TSJ3	*K 73 ADDR	22	24
TSM4	*K 64 ADDR	21	24
TSM5	*K 65 ADDR	21	24
TSM6	*K 66 ADDR	21	24

*Affected by relative mode.

<u>SICAP</u> <u>MNEMONIC</u>	<u>SICOM</u> <u>FORMAT</u>	<u>SICOM</u> <u>MANUAL PAGE #</u>	<u>SICOSAS</u> <u>APPENDIX A PAGE #</u>
TSR	K 70 ADDR	21	24
TZE	*K 60 ADDR	20	23
WRFILE	K 46 ADDR	12	22
XEQ	X 01 ADDR	34	32
ZEROIR	0 00 0010	25	26
*(Col. 1)			+

*Affected by relative mode.

APPENDIX C

Below is an alphabetized list of mnemonic operation codes and pseudo operation codes recognized by MLAP (Machine Language Assembly Program).

The referenced page number for the machine instructions is to Control Data 160-A Computer, Programming Manual, # 145A.

The * notation for the "relative" instructions refers the reader to page 27 of CDC Publication # 507 OSAS/A, The 160-A Assembly System.

The + notation for the pseudo operations recognized by MLAP refers the reader to the section entitled SICOSAS SPECIAL PURPOSE CARDS.

In the $Y \pm a$ notation, Y may be either symbolic or absolute. If absolute, Y will be considered either decimal or octal depending on the current mode of MLAP. (See MLAP card in Appendix D.) The additive field "a" may be used only if Y is symbolic and must be an absolute integer either octal or decimal again depending on the current mode of MLAP.

MANUAL #145: PAGE

ACJ	$Y \pm a$	Set direct, indirect, and relative bank control and jump.	36
ADB	$Y \pm a$	Add backward.	30
ADC	$Y \pm a$	Add constant.	30
ADD	$Y \pm a$	Add direct.	30
ADF	$Y \pm a$	Add forward.	30
ADI	$Y \pm a$	Add indirect.	30
ADM	$Y \pm a$	Add memory.	30
ADN	$Y \pm a$	Add no address.	30
ADR	$Y \pm a$	Add relative.	*
ADS		Add specific.	30
AOB	$Y \pm a$	Replace add one backward.	32
AOC	$Y \pm a$	Replace add one constant.	32
AOD	$Y \pm a$	Replace add one direct.	32
AOF	$Y \pm a$	Replace add one forward.	32
AOI	$Y \pm a$	Replace add one indirect.	32
AOM	$Y \pm a$	Replace add one memory.	32
AOR	$Y \pm a$	Replace add one relative.	*
AOS		Replace add one specific.	32

ATE	Y ± a	A to buffer entrance register.	25
ATX	Y ± a	A to buffer exit register.	25
BLS	Y ± a	Block store.	24
BSS		Block starting with symbol.	+
CBC		Clear buffer controls	39
CIL		Clear interrupt lockout.	40
CTA		Bank controls to A.	26
DRJ	Y ± a	Set direct and relative bank control and jump.	36
END		End compilation	+
EQU		Equality	+
ETA		Buffer entrance register to A	25
ERR		Error stop.	45
EXC	Y ± a	External function constant.	44
EXF	Y ± a	External function forward.	44
HLT		Halt	45
HWI	Y ± a	Half write indirect.	29
IBI	Y ± a	Initiate buffer input.	40
IBO	Y ± a	Initiate buffer output.	41
INA		Input to A.	43
IRJ	Y ± a	Set indirect and relative bank control and jump.	36
JFI	Y ± a	Jump forward indirect.	39
JPI	Y ± a	Jump indirect.	38
JPR	Y ± a	Return jump.	38
LCB	Y ± a	Load complement backward.	28
LCC	Y ± a	Load complement constant.	28
LCD	Y ± a	Load complement direct.	28
LCF	Y ± a	Load complement forward.	28
LCI	Y ± a	Load complement indirect.	28
LCM	Y ± a	Load complement memory.	28
LCN	Y ± a	Load complement no address.	28
LCR	Y ± a	Load complement relative.	*

LCS		Load complement specific.	28
LDB	Y ± a	Load backward.	27
LDC	Y ± a	Load constant.	27
LDD	Y ± a	Load direct.	27
LDF	Y ± a	Load forward.	27
LDI	Y ± a	Load indirect.	27
LDM	Y ± a	Load memory.	27
LDN	Y ± a	Load no address.	27
LDR	Y ± a	Load relative.	*
LDS		Load specific.	27
LPB	Y ± a	Logical product backward.	34
LPC	Y ± a	Logical product constant.	34
LPD	Y ± a	Logical product direct.	34
LPF	Y ± a	Logical product forward.	34
LPI	Y ± a	Logical product indirect.	34
LPM	Y ± a	Logical product memory.	34
LPN	Y ± a	Logical product no address.	34
LPR	Y ± a	Logical product relative	*
LPS		Logical product specific.	34
LSB	Y ± a (SCB)	Logical sum backward.	35
LSD	Y ± a (SCD)	Logical sum direct.	35
LSF	Y ± a (SCF)	Logical sum forward.	35
LSI	Y ± a (SCI)	Logical sum indirect.	35
LSN	Y ± a (SCN)	Logical sum no address.	35
LSR	Y ± a (SCR)	Logical sum relative.	*
LS1		Left shift one.	32
LS2		Left shift two.	32
LS3		Left shift three.	32
LS6		Left shift six.	32
MLAP		Set SICOSAS to MLAP mode.	+
MUH		Multiply A by one hundred.	30
MUT		Multiply A by ten.	29
NAME		(Name card)	+
NJB	Y ± a	Negative jump backward.	37

NJF	Y ± a	Negative jump forward.	37
NJR	Y ± a	Negative jump relative.	*
NOP	Y ± a	No operation.	45
NZB	Y ± a	Non-zero jump backward.	37
NZF	Y ± a	Non-zero jump forward.	37
NZR	Y ± a	Non-zero jump relative.	*
ORG		Origin	+
OTA		Output from A.	43
OTN	Y ± a	Output no address.	43
OUT	Y ± a	Output	41
PJB	Y ± a	Positive jump backward.	37
PJF	Y ± a	Positive jump forward.	37
PJR	Y ± a	Positive jump relative.	*
PTA		Transfer P to A.	24
RAB	Y ± a	Replace add backward.	31
RAC	Y ± a	Replace add constant.	31
RAD	Y ± a	Replace add direct.	31
RAF	Y ± a	Replace add forward.	31
RAI	Y ± a	Replace add indirect.	31
RAM	Y ± a	Replace add memory.	31
RAR	Y ± a	Replace add relative.	*
RAS		Replace add specific.	31
RETURN		Return to SICAP.	+
RS1		Right shift one.	32
RS2		Right shift two.	32
SICAP		Set SICOSAS to SICAP mode.	+
SSB	Y ± a	Subtract backward.	30
SBC	Y ± a	Subtract constant.	30
SBD	Y ± a	Subtract direct.	30
SBF	Y ± a	Subtract forward.	30
SBI	Y ± a	Subtract indirect.	30
SBM	Y ± a	Subtract memory.	30
SBN	Y ± a	Subtract no address.	30
SBR	Y ± a	Subtract relative.	*

SBS		Subtract specific.	30
SBU	Y \pm a	Set buffer bank control.	36
SCB	Y \pm a (LSB)	Selective complement backward.	35
SCC	Y \pm a	Selective complement constant.	35
SCD	Y \pm a (LSD)	Selective complement direct.	35
SCF	Y \pm a (LSF)	Selective complement forward.	35
SCI	Y \pm a (LSI)	Selective complement indirect.	35
SCM	Y \pm a	Selective complement memory.	35
SCN	Y \pm a (LSN)	Selective complement no address.	35
SCR	Y \pm a (LSR)	Selective complement relative.	*
SCS		Selective complement specific.	35
SDC	Y \pm a	Set direct bank control	36
SIC	Y \pm a	Set indirect bank control.	36
SID	Y \pm a	Set indirect and direct bank control.	36
SLJ	Y \pm a	Selective jump.	45
SLS	Y \pm a	Selective stop.	45
SJS	Y \pm a	Selective stop and jump.	45
SRB	Y \pm a	Shift replace backward.	33
SRC	Y \pm a	Shift replace constant.	33
SRD	Y \pm a	Shift replace direct.	33
SRF	Y \pm a	Shift replace forward.	33
SRI	Y \pm a	Shift replace indirect.	33
SRJ	Y \pm a	Set relative bank control and jump.	36
SRM	Y \pm a	Shift replace memory.	33
SRR	Y \pm a	Shift replace relative.	*
SRS		Shift replace specific	33
STB	Y \pm a	Store backward.	28
STC	Y \pm a	Store constant.	28
STD	Y \pm a	Store direct.	28
STE	Y \pm a	Store buffer entrance register at Location 6X and transfer A to buffer entrance register.	27
STF	Y \pm a	Store forward.	28
STI	Y \pm a	Store indirect.	28
STM	Y \pm a	Store memory.	28

STP	$Y \pm a$	Store P at location 5X.	26
STR	$Y \pm a$	Store relative.	*
STS		Store specific.	28
ZJB	$Y \pm a$	Zero jump backward.	37
ZJF	$Y \pm a$	Zero jump forward.	37
ZJR	$Y \pm a$	Zero jump relative.	*
*(Col. 1)		REMARKS Card.	+

DEC. 20, 1962
PAGE 1

* THIS IS A SAMPLE SICOMAS PROGRAM SHOWING SICAP COMMANDS ONLY
 * THE ORIGINAL PROGRAM CODED IN SICOM LANGUAGE APPEARS ON PAGE 42 OF
 * THE SICOM MANUAL.

43

THIS IS A 7090 TRANSFORMATION OF A SICOM PROGRAM WRITTEN FOR THE 160-A COMPUTER BY ANYONE U. WIS-
MLOC SILOC K OP ADDR SYMBOL OPCODE VARIABLE FIELD COMMENTARY
DEC.20,1962 PAGE 1

* THIS SAMPLE PROGRAM DEMONSTRATES COMMUNICATION BETWEEN A MAIN PROGRAM
* AND ITS SUBROUTINES, ONE OF WHICH (OUTPUT) IS IN MACHINE LANGUAGE.

LINE	ADDRESS	INSTR	OPERAND	COMMENT
3	0310	0100		
4	0310	0100		
5	0310	0101		
6	0314	0102		
7	0316	0103		
8	0320	0104		
9	0322	0105		
10	0324	0106		
11	0326	0107		
12	0330	0108		
13	0332	0109		
14	0334	0110		
15	0336	0111		
16	0340	0112		
17	0342	0113		
18	0344	0114		
19	0346	0115		
20	0350	0116		
21	0354	0118		
22	0360	0120		
23	0364	0122		
24	0366	0123		
25	0370	0124		
26	0374	0126		
27	0640	0208		
28	0644	0210		
29	0650	0212		
30				
31	0654	7500		
32	0655	4542		
33	0656	7301		
34	0657	0363		
35	0660	0100		

```

* THIS PROGRAM (LA07) ACCEPTS SYMBOLIC CODING IN BOTH A PSEUDO SICOM
* LANGUAGE AND 160-A MACHINE LANGUAGE AND PRODUCES A PROGRAM LISTING
* THE 160-A COMPUTER FOR EXECUTION
* PROGRAM TAPES...
* THIS PROGRAM (LA07) ACCEPTS SYMBOLIC CODING IN BOTH A PSEUDO SICOM
* LANGUAGE AND 160-A MACHINE LANGUAGE AND PRODUCES A PROGRAM LISTING
* AND A BINARY MAP OF THE PROGRAM ON MAGNETIC TAPE WHICH READS INTO
* THE 160-A COMPUTER FOR EXECUTION.
* PROGRAM TAPES...

```

```

*          A7 HI DENSITY..PROGRAM LISTING..PRINT PROGRAM CONTROL.
*          B7 HI DENSITY.. PROGRAM MEDIARY TAPE.
*          A8 HI DENSITY.. INPUT TAPE IF INPUT IS PRESTORED.
*          B8 LO DENSITY.. BINARY MAP OF ASSEMBLED PROGRAM.

```

```

          ORG      15200
START     HPR
          OTPEND   OTPEND
          +1 STL   22
          +4 SWT   1                CARD OR TAPE INPUT
          +5 TRA   SETOFF           TAPE
          +6 STZ   ONOFF           CARD
          +7 STZ   MCDE
          +8 CLA   CARD
          +9 STO   INUNIT
          +10 TRA  NOWIN
SETOFF    STL     ONOFF
          +1 CLA   INTAPE
          +2 STO   INUNIT
          IIMAGE   INPUT,,28
NOWIN     STL     43
          OIMAGE   OUTPUT,,28
          +4 STL   22
          IFILE    READ1
          +8 STL   43
          +11 AXT   20,4
          +12 CLA   NAME+20,4
          +13 STO   HEAD1+20,4
          +14 TIX   *-2,4,1
          ISCRIB   INUNIT
READ1     STL     43
          IFILE    READ2
          +4 STL   43
          +7 AXT   0,1                COUNT SYMBOLS
          ICHAR    ALLBNK,,2          NO SYMBOL
          +8 STL   43
          ICHAR    ISSYM,,13          NOT BLANK IN COLS 1-6
          +12 STL  43
          +16 CLA   ATE
          +17 STO   ICNTR
          ISCRIB   INUNIT
READ2     STL     43
          IBCW     WORK,,1,14
          +4 STL   43
          +8 STZ   TEMP
          OBCW     WORK,,1,14

```

APPENDIX F

+9	STL	22	
	CSCRIB	MEDTAP	
+13	STL	22	
	IBCC	TEMP,,1,1	WHAT KIND OF CARD
SKIP	STL	43	
+4	CLA	TEMP	TEST FOR COMMENT
+5	SUB	STAR	
+6	TZE	READ2	IS COMMEN IGNORE IT
	IBCW	CODE,,8,1	LOOK AT OPCODE
+7	STL	43	
+11	CLA	CCDE	
+12	CAS	SSQR	
+13	TRA	**2	
+14	TRA	FCRSQR	
+15	CAS	SLOG	
+16	TRA	**2	
+17	TRA	FCRLOG	
+18	CAS	SEXP	
+19	TRA	**2	
+20	TRA	FCREXP	
+21	CAS	SSIN	
+22	TRA	**2	
+23	TRA	FCRSIN	
+24	CAS	SATN	
+25	TRA	**2	
+26	TRA	FCRATN	
+27	CAS	SICOM	
+28	TRA	CASGOB	
+29	TRA	**2	IS SICOM CARD
+30	TRA	CASGOB	
+31	STZ	MCDE	
+32	TRA	READ2	
CASGOB	CAS	GCBACK	CHECK RETURN CARD
+1	TRA	CAS160	
+2	TRA	**2	IS IGNORE FIRST
+3	TRA	CAS160	
+4	CLA	ICNTR	
+5	ADD	TWO	
+6	STO	ICNTR	
+7	TRA	ISOP	
CAS160	CAS	160A	MACHINE MODE
+1	TRA	CASNAM	
+2	TRA	**2	
+3	TRA	CASNAM	
+4	STL	MCDE	
+5	TRA	READ2	
CASNAM	CAS	NAMER	
+1	TRA	CASORG	
+2	TRA	**2	
+3	TRA	CASCRG	
	IBCW	HEAD1+14,,16,6	
+4	STL	43	
+8	TRA	READ2	
CASCRG	CAS	ORIGIN	

+1	TRA	**2	
+2	TRA	ISORG	
+3	CAS	DECDAT	
+4	TRA	**2	
+5	TRA	ISDEC	
+6	CAS	EXTMSK	
+7	TRA	**2	
+8	TRA	ISDEC	
+9	CAS	END	
+10	TRA	**2	
+11	TRA	ISEND	
+12	CAS	EQU	
+13	TRA	**2	
+14	TRA	READ2	
+15	CAS	BSS	
+16	TRA	**2	
+17	TRA	ISBSS	
+18	NZT	MCDE	
+19	TRA	ISOP	
+20	AXT	BTYPE-MINSTR,4	
CASBTY	CAS	BTYPE,4	
+1	TRA	**2	
+2	TRA	LITEIT	
+3	TIX	CASBTY,4,1	
+4	TRA	ISOP	
LITEIT	SLN	1	
	IINT	TEMP+1,,1,6	
ISCP	STL	43	LOOK FOR SYMBOL
ALLBNK	CLA	ICNTR	MACHINE LOCATION COUNTER
+1	ADD	TWO	
+2	ZET	MCDE	
+3	SUB	ONE	
+4	STO	ICNTR	
+5	SLT	1	WAS THIS DEC DATA CARD
+6	TRA	READ2	NO
+7	TRA	ALLBNK+1	
	IBCW	SYMTAB,1,1,1	
ISSYM	STL	43	
+4	CLA	ICNTR	
+5	STO	LCCTAB,1	
+6	TXI	**1,1,-1	
+7	SXA	SYMCNT,1	
+8	CLA	ICNTR	-SYMBOL COUNT
AGN	ADD	TWO	
+1	ZET	MCDE	
+2	SUB	ONE	
+3	STO	ICNTR	
+4	SLT	1	
+5	TRA	READ2	
+6	TRA	AGN	WAS DEC DATA
	IBCW	SYMTAB,1,1,1	
ISBSS	STL	43	
+4	CLA	ICNTR	
+5	STO	LCCTAB,1	

+6	TXI	*+1,1,-1
+7	SXA	SYMCNT,1
	IINT	TEMP+3,,16
+8	STL	43
+12	CLA	TEMP+3
+13	ZET	MODE
+14	TRA	*+2
+15	ALS	1
+16	ADD	ICNTR
+17	STO	ICNTR
+18	TRA	READ2
	IINT	ICNTR,,16
ISORG	STL	43
+4	CLA	ICNTR
+5	ALS	1
+6	STO	ICNTR
+7	TRA	READ2
ISDEC	ZET	MODE
+1	TRA	ISOP
+2	SLN	1
+3	TSX	EVENUP,4
+4	TRA	ISOP
	IBCW	DYNAMO,,16,1
ISEND	STL	43
+4	REWB	7
+5	TSX	FNDSYM,4
+6	TNZ	*+2
+7	CLA	LCCTAB
+8	ARS	1
+9	STA	GC2
+10	CLA	ATE
+11	STO	ICNTR
	CBLANK	120,,1
+12	STL	22
	CREADY	
+15	STL	22
	CBLANK	120,,1
+17	STL	22
	CEOR	DUMP,,132
+20	STL	22
	IFILE	NOWGO
+24	STL	43
+27	AXT	4096,4
+28	STZ	SIMAP+4096,4
+29	TIX	*-1,4,1
+30	AXT	56,4
+31	CLA	BLANK
+32	STO	INPUT+56,4
+33	TIX	*-1,4,1
+34	WPDA	
+35	SPRA	1
	CHEAD	CASE1,,1,,1
+36	STL	22
+41	STZ	PAGENC

MAKE AN EVEN MACHINE LOCATION
2XSICOM LOC = MACHINE LOCATION

+42	STZ	MCDE
IMAGE1	EQU	CUTPUT+21
IMAGE2	EQU	IMAGE1+28
	ICHAR	0
-4	STL	43
	ISCRIB	MEDTAP
NOWGO	STL	43
	IREADY	
+4	STL	43
	IBCW	TEMP,,8,1
+6	STL	43
+10	CLA	TEMP
+11	SUB	BLANK
+12	TZE	NOWGO
	IREADY	
+13	STL	43
	IFILE	CYCLE
+15	STL	43
+18	CLA	ONE
+19	STO	NUMBER
	ISCRIB	MEDTAP
CYCLE	STL	43
	IBCW	WORK,,1,12
+4	STL	43
	OBCW	WORK,,28,12
+8	STL	22
	OINT	NUMBER,,115,4
+12	STL	22
+16	CLA	NUMBER
+17	ADD	ONE
+18	STO	NUMBER
+19	STZ	TEMP
	IBCC	TEMP,,1,1
+20	STL	43
+24	CLA	TEMP
+25	SUB	STAR
+26	TZE	CMTCRD
	IBCW	CODE,,8,1
+27	STL	43
+31	CLA	CODE
+32	CAS	SSQR
+33	TRA	**2
+34	TRA	SCRFOR
+35	CAS	SLOG
+36	TRA	**2
+37	TRA	LCGFOR
+38	CAS	SEXP
+39	TRA	**2
+40	TRA	EXPFOR
+41	CAS	SSIN
+42	TRA	**2
+43	TRA	SINFOR
+44	CAS	SATN
+45	TRA	**2

COLS 121-126 OF OUTRAN REGIONS

TEST FOR COMMENT CARD

FOUND ONE
TEST NOW FOR

	+46	TRA	ATNFOR	
	+47	CAS	SICCM	
	+48	TRA	Z1	
	+49	TRA	*+2	
	+50	TRA	Z1	
	+51	STZ	MCDE	
	+52	STZ	IMODE	
	+53	TRA	CMTCRD	
Z1		CAS	GCBACK	
	+1	TRA	Z2	
	+2	TRA	FCRART	FOUND A RETURN CARD
Z2		CAS	160A	
	+1	TRA	Z3	
	+2	TRA	*+2	
	+3	TRA	Z3	
	+4	STL	MODE	
		ICHAR	Ø	
	+5	STL	43	
		ICHAR	YISOCT,,8	
	+9	STL	43	
		IINT	NEWTEM,,16,6	
	+13	STL	43	
	+17	STZ	IMODE	
	+18	TRA	CMTCRD	
		ICHAR	Ø	
YISOCT		STL	43	
	+4	STL	IMODE	
	+5	TRA	CMTCRD	
Z3		CAS	OCTAL	
	+1	TRA	Z4	
	+2	TRA	CCTCRD	
Z4		CAS	BLANK	
	+1	TRA	Z5	
	+2	TRA	BNKCRD	
Z5		CAS	NAMER	
	+1	TRA	*+2	
	+2	TRA	CYCLEX	
	+3	CAS	CRIGIN	ORIGIN CARD
	+4	TRA	*+2	
	+5	TRA	CRGCRD	
	+6	CAS	DECDAT	DEC DATA CARD
	+7	TRA	*+2	
	+8	TRA	DATCRD	
	+9	CAS	EXTMSK	
	+10	TRA	BSSCAS	
	+11	TRA	*+2	
	+12	TRA	BSSCAS	
	+13	STL	MASKER	
	+14	TRA	DATCRD	
CYCLEX		CLA	NUMBER	
	+1	SUB	ONE	
	+2	STO	NUMBER	
	+3	TRA	CYCLE	
BSSCAS		CAS	BSS	

+1	TRA	#+2	
+2	TRA	BSSCRD	
+3	CAS	EQU	EQU CARD
+4	TRA	#+2	
+5	TRA	EQUCRD	
+6	CAS	END	END CARD
+7	TRA	#+2	
+8	TRA	ENDCRD	
+9	NZT	MODE	
+10	TRA	STABLE	
+11	AXT	LSTINS-MINSTR,1	IS MACHINE CODE
GOLOOK	CAS	LSTINS,1	
+1	TRA	#+2	
+2	TRA	THANX	
+3	TIX	GOLOOK,1,1	
+4	TRA	BADCOD	
THANX	CLA	LSTINS,1	
+1	STO	RITEOP	
+2	CLA	LSTKEY,1	
+3	STO	RITEKY	
+4	LAC	RITEKY,2	
+5	TRA*	AFORK,2	
AFORK	PZE	0TYPE	EG BLS 0100 YYYY
+1	PZE	1TYPE	EG PTA 0101
+2	PZE	2TYPE	EG STP 015X
+3	PZE	3TYPE	EG LDN 04XX
+4	PZE	4TYPE	EG SLJ 77X0
+5	PZE	5TYPE	RELATIVE
STABLE	AXT	ENDOP-OPCODE,1	
CMPCOD	CAS	ENDOP,1	FIND CORRECT CODE IN TABLE
+1	TRA	#+2	
+2	TRA	FNDOP	MATCHED
+3	TIX	CMPCOD,1,1	
+4	TRA	BADCOD	
FNDOP	CLA	ENDOP,1	
+1	STO	RITEOP	
+2	CLA	ENDKEY,1	
+3	STO	RITEKY	
+4	LAC	RITEKY,2	
+5	TRA*	FORK,2	
FORK	PZE	TYPE0	KOPADDR E.G. CLA MPY ETC
+1	PZE	TYPE1	OP=DR E.G. NOP TMK5
+2	PZE	TYPE2	K D18 SEL MAG TAPE
+3	PZE	TYPE3	K DR E.G. BLOCK COPY
+4	PZE	TYPE4	AD19 BACKSPACE
+5	PZE	TYPE5	K00AD2R PXA PAX
+6	PZE	TYPE6	KOP=OP EG CLAC GO ETC
	ICHAR	YESSYM,,8	
TYPE0	STL	43	
	IINT	SIADDR,,16,6	
+4	STL	43	
	ICHAR	0	
+8	STL	43	
+12	TSX	VARFRM,4	

+13	NOP		
+14	TRA	NOPE	
	ICHAR	Ø	
YESSYM	STL	43	
+4	TSX	FNDSYM,4	
+5	ARS	1	
+6	STO	SIADDR	
+7	SLT	2	WAS IT AC SYMBOL
+8	TRA	NOPE	NO
+9	CAL	ACLIST	\$ØØØØØ
+1Ø	ORS	RITEKY	
+11	SLN	2	FOR BINARY ROUTINE
+12	TRA	TSXS	
NOPE	CAL	INDEXR	
+1	ORS	RITEKY	
TSXS	TSX	STOKOP,4	
ALMSTØ	TSX	CNV SAD,4	SET UP HOLLERITH ADDR
ASIFØ	TSX	OKTODR,4	MOVE K OP ADDR TO OUTRAN OUTPUT REGI1Ø
+1	TSX	LCCMCH,4	PUT MACHINE LOCATION IN OUTRAN REGION
+2	TSX	LCCSIC,4	PUT SICOM LOCATION IN OUTPUT REGION
+3	TSX	WRLIST,4	WRITE ONE LINE ON LISTING TAPE
+4	SLT	3	
+5	TRA	GCAHED	
+6	CLA	RETURN	
+7	ADD	ONE	
+8	STA	**1	
+9	TRA	**Ø	
GOAHED	TSX	STOCMD,4	
REBACK	CLA	ICNTR	
+1	ADD	TWO	
+2	ZET	MODE	
+3	SUB	ONE	
+4	STO	ICNTR	
+5	SLT	1	
+6	TRA	CYCLE	
+7	TRA	REBACK+1	
TYPE1	AXT	5,4	ZERO K O P A D FOR TYPE 1 INSTR
+1	STZ	K+5,4	
+2	TIX	*-1,4,1	
+3	CLA	ZERO	
+4	LDQ	RITEKY	
+5	RQL	6	
+6	LGL	6	
+7	STO	D2	
+8	CLA	ZERO	
+9	LGL	6	
+1Ø	STO	R	
+11	TSX	OKTODR,4	MOVE ØØØØØDR TO OUTPUT REGION
+12	TSX	LOCMCH,4	
+13	TSX	LCCSIC,4	
+14	TSX	WRLIST,4	
+15	LDQ	D2	
+16	MPY	DECIMAL+2	
+17	XCA		

+18	ALI	R
+19	STO	SIADDR
+20	TSX	STCCMD,4
+21	TRA	REBACK
TYPE2	STZ	G
+1	STZ	P
+2	STZ	A
+3	CLA	ONE
+4	STC	D2
+5	CLA	ATE
+6	STO	R
+7	TSX	VARFRM,4
+8	NOP	

DONE THIS CARD
SEL MAC TAPE

	IMASK	0
+9	STL	43
+13	LDQ	D1
+14	MPY	DECIMAL+1
+15	STQ	TEMP
+16	LDQ	D2
+17	MPY	DECIMAL+2
+18	XCA	

+19	ADD	R
+20	ADD	TEMP
+21	STO	SIADDR
+22	TRA	ASIF0
TYPE3	TSX	WIPE,4
+1	TSX	VARFRM,4
+2	NOP	

+3	LDQ	RITEKY
+4	CLA	ZERC
+5	RQL	6
+6	LGL	6
+7	STC	D2
+8	CLA	ZERC
+9	LGL	6
+10	STO	R
+11	LDQ	D2
+12	MPY	DECIMAL+2
+13	XCA	
+14	ADD	R
+15	STC	SIADDR
+16	TRA	ASIF0

TYPE4	TSX	WIPE,4
+1	CLA	ONE
+2	STO	D2
+3	CLA	NINE
+4	STO	R

BACKSPACE AD19

	IINT	AD,,16
+5	STL	43
+9	LDQ	AC
+10	CLA	ZERC
+11	DVH	DECIMAL+2
+12	STQ	A
+13	STO	D1

	+14	XCA	
	+15	ALS	3
	+16	STO	K
	+17	LDQ	D1
	+18	MPY	DECMAL+1
	+19	XCA	
	+20	ADD	D19
	+21	STO	SIADDR
	+22	TRA	ASIF0
TYPE5	TSX	WIPE,4	PAX,PXA 22 OR 23
	+1	TSX	VARFRM,4
	+2	NOP	
	+3	CLA	TW0
	+4	STO	D2
	+5	LDQ	RITEKY
	+6	CLA	ZERO
	+7	LGL	6
	+8	STO	A
	+9	CLA	ZERO
	+10	LGL	6
	+11	STO	D1
	+12	CLA	ZERO
	+13	LGL	6
	+14	STO	R
	+15	AXT	3,4
LDQA	LDQ	A+3,4	
	+1	MPY	DECMAL+3,4
	+2	STQ	TEMP+3,4
	+3	TIX	LDQA,4,1
	+4	CLA	R
	+5	ADD	TEMP
	+6	ADD	TEMP+1
	+7	ADD	TEMP+2
	+8	STO	SIADDR
	+9	TRA	ASIF0
TYPE6	TSX	WIPE,4	KOP = OP ADDR
	+1	AXT	3,4
	+2	LDQ	RITEKY
LOOP	CLA	ZERO	
	+1	LGL	6
	+2	STO	K+3,4
	+3	TIX	LOOP,4,1
		ICHAR	NOTK,,8
	+4	STL	43
		IINT	SIADDR,,16,6
	+8	STL	43
		ICHAR	0
	+12	STL	43
	+16	TRA	SEEIF
NOTK	TSX	FNDSYM,4	ADDR IS SYMBOLIC
	+1	ARS	1
	+2	STO	SIADDR
		ICHAR	0
	+3	STL	43

```

SEEIF  CLA  SIADDR
+1  CAS  D1999
+2  TRA  FIXIT
+3  TRA  ALMST0
+4  TRA  ALMST0
FIXIT  SLN  3
+1  STL  RETURN
+2  TRA  ALMST0
BACKIN LDQ  SIADDR
+1  CLA  ZERO
+2  DVH  D2000
+3  STO  SIADDR
+4  STQ  CVRAGE
+5  XCA
+6  LRS  2
+7  LBT
+8  TRA  NCT8
+9  CLA  8IND
+10 ORS  SIADDR
+11 TRA  GCAHED
NCT8  LLS  2
+1  ALS  4
+2  ORS  K
+3  TRA  GCAHED
      BEGIN  1,4
WIPE  TXL  **3,0,0
+4  AXT  7,4
+5  STZ  K+7,4
+6  TIX  *-1,4,1
      RETURN WIPE
+7  TRA  WIPE+1
      BEGIN  1,4
STCKCP TXL  **3,0,0
+4  LDQ  RITEKY
+5  AXT  3,4
CLAZ  CLA  ZERC
+1  LGL  6
+2  STO  K+3,4
+3  TIX  CLAZ,4,1
      RETURN STCKCP
+4  TRA  STCKCP+1
      BEGIN  1,4
CNVSAD TXL  **3,0,0
+4  AXT  3,4
+5  LDQ  SIADDR
NXTONE CLA  ZERC
+1  DVH  DECIMAL+3,4
+2  STQ  A+3,4
+3  XCA
+4  TIX  NXTONE,4,1
+5  STQ  R
      RETURN CNVSAD
+6  TRA  CNVSAD+1
      BEGIN  1,4

```

2000 OR LESS
NUMBER OF 2000 NDS

TEST FOR 8000

IS 8000
4000

SUBROUTINE LINKAGE

SET UP HOLLERITH K,O,P
SUBROUTINE LINKAGE

SUBROUTINE LINKAGE
CONVER BINARY SICOM ADDR TO BCD

```

CKTCDR  TXL      *+3,0,0
        OMASK    6,,31
        +4  STL      22
        OBCC     K,,15,1
        +8  STL      22
        OBCC     C,,17,1
        +12 STL      22
        OBCC     P,,18,1
        +16 STL      22
        +20 AXT      4,4
        CCOLR    20
        +21 STL      22
        OBCC     A+4,4,,1
CUTATR  STL      22
        +4  TIX      OUTATR,4,1
        OMASK    0
        +5  STL      22
        RETURN   CKTCDR
        +9  TRA      OKTODR+1
IREDUN  HPR      1
        +1  TRA      START
CREDUN  HPR      2
        +1  TRA      START
CTPEND  HPR      3
        +1  TRA      WRLIST
NOEND   HPR      4
        +1  TRA      ENDCRD
        BEGIN    1,4
LOCMCH  TXL      *+3,0,0
        +4  CLA      ICNTR
        +5  ADD      OCTGND
        +6  STO      TEMP+4
        OOCTAL    TEMP+4,,2,5
        +7  STL      22
        OBCC     BLANK,,2,1
        +11 STL      22
        RETURN   LOCMCH
        +15 TRA      LOCMCH+1
        BEGIN    1,4
LOCSIC  TXL      *+3,0,0
        +4  CLA      ICNTR
        +5  ARS      1
        +6  ADD      10THOU
        +7  STO      TEMP+3
        CINT     TEMP+3,,7,5
        +8  STL      22
        OBCC     BLANK,,8,1
        +12 STL      22
        RETURN   LOCSIC
        +16 TRA      LOCSIC+1
        BEGIN    1,4
WRLIST  TXL      *+3,0,0
        CSCRIB    CUTAPE,,20,1
        +4  STL      22

```

SUBROUTINE LINKAGE

SUBROUTINE LINKAGE
CONVER LOCATION COUNTER IN OUTPUT REGIO
OCTAL 10000

SUBROUTINE LINKAGE
DECIMAL 10000

SUBROUTINE LINKAGE

	+8	SWT	3
	+9	TRA	OBLANK
		OREADY	
	+10	STL	22
		OSCRIB	PRINTR,,20
	+12	STL	22
		OBLANK	120,,1
OBLANK		STL	22
		RETURN	WRLIST
	+3	TRA	WRLIST+1
CMTCRD		TSX	WRLIST,4
	+1	TRA	CYCLE
		IINT	ICNTR,,16
ORGCRD		STL	43
	+4	CLA	ICNTR
	+5	ALS	1
	+6	STO	ICNTR
	+7	TSX	LOCMCH,4
	+8	TSX	LOCSIC,4
	+9	TSX	WRLIST,4
	+10	TRA	CYCLE
BSSCRD		TSX	LOCMCH,4
	+1	NZT	MODE
	+2	TSX	LOCSIC,4
	+3	TSX	WRLIST,4
		IINT	NEWTEM,,16
	+4	STL	43
	+8	CLA	NEWTEM
	+9	NZT	MODE
	+10	ALS	1
	+11	ADD	ICNTR
	+12	STO	ICNTR
	+13	TRA	CYCLE
		ICHAR	SYMBOL,,8
EQUCRD		STL	43
	+4	LXA	SYMCNT,1
		IBCW	SYMTAB,1,1,1
	+5	STL	43
		IMASK	0
	+9	STL	43
		IINT	NEWTEM+1,,16,6
	+13	STL	43
	+17	CLA	NEWTEM+1
	+18	ZET	MODE
	+19	TRA	FORMCH
	+20	ALS	1
	+21	STO	LOCTAB,1
	+22	ARS	1
REVRS		TXI	*+1,1,-1
	+1	SXA	SYMCNT,1
	+2	ADD	10THOU
	+3	STO	NEWTEM+2
		OINT	NEWTEM+2,,18,5
	+4	STL	22

FOUND A COMMENT CARD ON 2ND PASS

```

      OBCC      BLANK,,19,1
+8    STL      22
      ICHAR     0
+12   STL      43
+16   TSX      WRLIST,4
+17   TRA      CYCLE
FORMCH STO      LCCTAB,1
      COCTAL    LOCTAB,1,20,4
+1    STL      22
+5    TXI      **1,1,-1
+6    SXA      SYMCNT,1
      ICHAR     0
+7    STL      43
+11   TSX      WRLIST,4
+12   TRA      CYCLE
SYMBCL TSX      FNDSYM,4
+1    ZET      MCDE
+2    TRA      FORMCH
+3    STO      LCCTAB,1
+4    ARS      1
+5    TRA      REVRS
DATCRD ZET      MCDE
+1    TRA      BNKCRD
+2    SLN      1
+3    TSX      EVENUP,4
+4    TSX      SCANIT,4
+5    TSX      LCCMCH,4
+6    TSX      LCCSIC,4
+7    TSX      WRLIST,4
+8    TRA      REBACK
ENDCRD CLA      GC2
+1    ADD      10THOU
+2    STO      TEMP
+3    REWB     7
      OINT      TEMP,,18,5
+4    STL      22
      OBCC      BLANK,,19,1
+8    STL      22
+12   TSX      WRLIST,4
+13   WEFA     7
+14   TSX      RITOUT,4
+15   SWT      6
+16   TRA      **2
+17   TRA      **2
+18   REWA     7
      OBCW      ALLDUN,,1,20
+19   STL      22
      CSCRI8    PRINTR,,20
+23   STL      22
+27   WPDA
+28   SPRA     1
+29   HTR      START+1
      BEGIN     1,4
BADCCD TXL      **3,0,0

```

WRITE THIS DEC ISCAN ROUTINE

SUBROUTINE LINKAGE

```

+4  STO      NOCODE
    OREADY
+5  STL      22
    OBCW     NOCODE,,1,20
+7  STL      22
+11 TSX      ERROR,4
+12 AXT      0,1
+13 ZET      MCDE
+14 TRA      THANX
+15 TRA      FNDOP
* THIS ROUTINE WILL SET UP TWO REGISTERS COMPRISING A SICOM COMMAND
  BEGIN      1,7
STOCMD TXL    **5,0,0          SUBROUTINE LINKAGE
+8  CLA      WHRMAP
+9  ADD      ICNTR
+10 STA      STOREV          EVEN LOCATION = SIADDR
+11 ADD      ONE
+12 STA      STOROD          ODD LOCATION = KOP
+13 CLA      SIADDR
STOREV STO    **0
+1  SLT      2              IS AC
+2  TRA      NOTAC          NO
+3  CLA      ABLNK          YES      00000B
ALS3   ALS    3
+1  ORA      0
+2  ALS      3
+3  ORA      P
STOROD STO    **0
      RETURN STOCMD
+1  TRA      STOCMD+1
NOTAC  CLA      RITEKY
+1  TMI      **3          IS XEQ,SETFWA,SETLWA
+2  CLA      K
+3  TRA      ALS3
+4  SSP
+5  ARS      30
+6  SUB      OCT12
+7  TRA      ALS3
OCT12  OCT      12
      BEGIN      1,7
FNDSYM TXL    **5,0,0          SUBROUTINE LINKAGE
+8  TSX      VARFRM,4
+9  TRA      **2          SPECIAL RETURN
+10 TRA      NORMAL
+11 CLA      MYOWN
+12 ALS      1
      RETURN FNDSYM
+13 TRA      FNDSYM+1
NORMAL AXT      0,4
+1  LAC      SYMCNT,1      NUMBER SYMBOLS STORED THUS FAR
+2  CLA      AFIELD
+3  CAS      ACCODE
+4  TRA      **2          CHECK FOR AC = SIADDR
+5  TRA      ISAC

```


LCCKAT	CAS	SYMTAB,4	
+1	TRA	*+2	
+2	TRA	FCUND	
+3	TXI	*+1,4,-1	
+4	TIX	LCCKAT,1,1	
+5	STO	UNDEF	SYMBOL IS UNDEFINED
	OREADY		
+6	STL	22	
	OBCW	UNDEF,,1,20	
+8	STL	22	
+12	TSX	ERROR,4	
+13	CLA	ZERO	
+14	TRA	FOUNDX	
ISAC	SLN	2	
+1	CLA	ACNUM	
	RETURN	FNDSYM	
+2	TRA	FNDSYM+1	
FOUND	CLA	LOCTAB,4	
+1	STO	EQUSYM	
+2	CLA	AFIELD	
+3	TXI	*+1,4,-1	
+4	TNX	FINE,1,1	
CMPSYM	CAS	SYMTAB,4	
+1	TRA	*+2	
+2	TRA	STUPID	
+3	TXI	*+1,4,-1	
+4	TIX	CMPSYM,1,1	
FINE	CLA	EQUSYM	
+1	ADD	ADDIT	
	RETURN	FNDSYM	
FOUNDX	TRA	FNDSYM+1	
STUPID	STO	DOUBLE	
	OREADY		
+1	STL	22	
	OBCW	DOUBLE,,1,20	
+3	STL	22	
+7	TSX	ERROR,4	
+8	TRA	FINE	
	BEGIN	1,4	
ERROR	TXL	*+3,0,0	SUBROUTINE LINKAGE
	OSCRIB	OUTAPE,,20,1	
+4	STL	22	
+8	SWT	3	
	RETURN	ERROR	
+9	TRA	ERROR+1	
	OREADY		
+10	STL	22	
	OSCRIB	PRINTR,,20	
+12	STL	22	
	RETURN	ERROR	
+16	TRA	ERROR+1	
	BEGIN	2,7	
VARFRM	TXL	*+5,0,0	SUBROUTINE LINKAGE
	IMASK	,,31	

+8	STL	43	
+12	AXT	15,4	
	ICOLR	16	
+13	STL	43	
RAVEL	STZ	DATA+15,4	
	IBCC	DATA+15,4,,1	
+1	STL	43	
+5	TIX	RAVEL,4,1	
	IMASK	0	COLS 16-30 UNPACKED
+6	STL	43	
+10	STZ	CCMCOL	0=COL 16
+11	STZ	ADDIT	
+12	STZ	ADDCOL	
+13	STZ	INDEXR	
+14	AXT	15,4	
+15	AXT	0,1	
+16	CLA	ISBLNK	LOOK FOR BLANK
LOG1	CAS	DATA+15,4	
+1	TRA	**2	
+2	TRA	FNDBNK	FOUND BLANK
+3	TXI	**1,1,1	
+4	TIX	LOG1,4,1	
+5	AXT	0,1	
FNDBNK	SXA	BNKCOL,1	COLUMN NO-16 OF BLANK
+1	LXA	BNKCOL,4	
+2	CLA	BNKCOL	
+3	ADD	WHRDAT	
+4	STA	REFADD	
+5	AXT	0,1	
LOG2	CLA	ISPOS	LOOK FOR ADDITIVE SYMBOL
+1	CAS*	REFADD	
+2	TRA	**2	
+3	TRA	FNDADD	
+4	CLA	ISNEG	
+5	CAS*	REFADD	
+6	TRA	**2	
+7	TRA	FNDADD	
+8	TXI	**1,1,1	
+9	TIX	LOG2,4,1	
+10	AXT	0,1	
FNDADD	SXA	ADDCOL,1	COL NO (-16) OF ADDITIVE SYMBOL
+1	LXA	BNKCOL,4	
+2	AXT	0,1	
+3	CLA	ISCOM	LOOK FOR COMMA
LOG3	CAS*	REFADD	
+1	TRA	**2	
+2	TRA	FNDCOM	
+3	TXI	**1,1,1	
+4	TIX	LOG3,4,1	
+5	AXT	0,1	
FNDCOM	SXA	CCMCOL,1	COMMA IN THIS COLUMN
+1	ZET	COMCOL	
+2	TRA	YSCOMA	WAS A COMMA
ZETADD	ZET	ADDCOL	NO COMMA

+1	TRA	YESADD	WAS AN ADDITIVE FIELD
	IBCW	AFIELD,,16,1	SYMBOL AS IT STANDS IS OK
+2	STL	43	
	RETURN	VARFRM	NO ADD NO COMMA
+6	TRA	VARFRM+1	
YESADD	CLA	SIXX	= MAYBE COMMA BUT ADDITIVE FIRST
+1	SUB	ADDCOL	ISCLATE SYMBOL FROM + OR -
INTPL	TPL	*+2	
+1	CLA	ZERC	ERROR
+2	XCA		
+3	MPY	SIXX	
+4	XCA		
+5	STA	SHFTR	(6-C)X6 = NO BITS TO RIGHT OF SYMEOL
+6	LXA	ADDCOL,1	NO CHARACTERS IN SYMBOL
+7	AXT	0,2	
+8	CLA	ZERC	
ROTAT	ALS	6	
+1	ORA	DATA,2	
+2	TXI	*+1,2,-1	
+3	TIX	RCTAT,1,1	
+4	LDQ	BLANK	6 BCD BLANKS
SHFTR	LGL	**0	
+1	SLW	AFIELD	SYMBOL UNDRESSED STORED FOR LOOK UP
+2	SLT	2	
+3	TRA	*+2	
+4	TRA	ISTAG	
+5	CLA	ADDCOL	NOW GET ADDITIVE FIELD
+6	ADD	CCLDIF	
+7	STA	IFDEC+3	I MACROS
+8	STA	IFOCT+3	
+9	CLA	CCMCOL	
+10	ZET	CCMCOL	
+11	TRA	*+2	
+12	CLA	BNKCOL	
+13	SUB	ADDCOL	
+14	ALS	18	
+15	STD	IFDEC+3	
+16	STD	IFOCT+3	
+17	ZET	IMODE	ZERO = DEC MODE
+18	TRA	ACKOCT	
	ICHAR	NONOCT,,8	
+19	STL	43	
	IINT	ADDIT,,**,**	
IFDEC	STL	43	
+4	ZET	MODE	
+5	TRA	MOVEON	
+6	CLA	ADDIT	
+7	ALS	1	
+8	STO	ADDIT	
MOVEON	NZT	COMCOL	IS THERE A TAG FIELD
	RETURN	VARFRM	SYMBOL SET IN AFIELD +OR- IN ADDIT
+1	TRA	VARFRM+1	
+2	TRA	ISTAG	
	ICHAR	NONOCT,,8	

ACKOCT	STL	43
	IOCTAL	ADDIT,,**,**
IFOCT	STL	43
	RETURN	VARFRM
+4	TRA	VARFRM+1
	ICHAR	Ø
NONOCT	STL	43
	OREADY	
+4	STL	22
	OBCW	BADVAR,,1,2Ø
+6	STL	22
+1Ø	TSX	ERROR,4
+11	STZ	ADDIT
	RETURN	VARFRM
+12	TRA	VARFRM+1
YSCOMA	ZET	ADDCOL
+1	TRA	YESADD
+2	TRA	HARD1
ISTAG	CLA	COMCOL
+1	ADD	COLDIF
+2	ADD	ONE
+3	STA	PROTAG+3
	IINT	K,,**,1
PROTAG	STL	43
+4	CLA	K
+5	ALS	3Ø
+6	SLW	INDEXR
+7	SLT	1
	RETURN	VARFRM
+8	TRA	VARFRM+1
	RETURN	VARFRM,1
+9	AXT	1,4
	ICHAR	Ø
HARD1	STL	43
	ICHAR	ISALPH,,8
+4	STL	43
+8	CLA	COMCOL
+9	ALS	18
+1Ø	STD	TRYIT+3
	IINT	MYOWN,,16,**
TRYIT	STL	43
+4	SLN	1
+5	TRA	ISTAG
	ICHAR	Ø
ISALPH	STL	43
+4	SLN	2
+5	CLA	COMCOL
+6	STO	ADDCOL
+7	CLA	SIXX
+8	SUB	COMCOL
+9	TRA	INTPL
COMCOL	PZE	
BNKCOL	PZE	
ADDCOL	PZE	

IS THERE ADDITIVE BESIDED TAG
 PROCES TAG FIELD LATER
 FIRST FIELD MAY NUMERIC

THIS CASE HAS NO ADD BUT TAG + MAYBE N

ISCOM	BCI	1,00000,
INDEXR	PZE	
SIXX	DEC	6
ADDIT	PZE	
COLDIF	DEC	16
MYCWN	PZE	

	BEGIN	1,7
SCANIT	TXL	*+5,0,0
+8	CLA	WHRMAP
+9	ADD	ICNTR
+10	STA	D890
+11	ADD	ONE
+12	STA	D567
+13	ADD	ONE
+14	STA	D234
+15	ADD	ONE
+16	STA	EXPDI
	IMASK	,,31
+17	STL	43
+21	AXT	15,4
	ICOLR	16
+22	STL	43
UNPACK	STZ	DATA+15,4
	IBCC	DATA+15,4,,1
+1	STL	43
+5	TIX	UNPACK,4,1
	IMASK	0
+6	STL	43
+10	CLA	EXP0
+11	STO	EXPCNT
+12	AXT	10,4
+13	STZ	DIGIT+10,4
+14	TIX	*-1,4,1
+15	STZ	MANSNG
+16	STZ	EPART
+17	STZ	1STNZ
+18	STZ	DPTIND
+19	AXT	15,1
+20	AXT	0,2
+21	AXT	0,4
+22	STZ	DPTIND
LCKKC	CLA	DATA+15,1
+1	CAS	ISBLNK
+2	TRA	*+2
+3	TRA	GETCUT
+4	CAS	TEN
+5	TRA	GR10
+6	HPR	
+7	TNZ	MARKIT
+8	ZET	1STNZ
+9	TRA	*+2
+10	TRA	MCRCLM
MARKIT	NZT	1STNZ
+1	TRA	1STCNE

SUBROUTINE LINKAGE
ASSEMBLE + STORE DEC CARD COLS 16+

UNPACK 15 COLS STARTING COL 16

OCT 100

LOOK AT 15 COLUMNS
COLUMN COUNTER
DIGIT STORER

LOOK AT A COLUMN

FOUND A BLANK

ERROR
GR0 IS BETWEEN 1 AND 9
IS THIS A LEADING 0
NO
YES IGNORE IT

STODIG	STO	DIGIT,4	
+1	TXI	*+1,4,-1	
MORCLM	TXI	*+1,2,1	
+1	TIX	LCOKC,1,1	
+2	TRA	ILEGAL	
GR10	CAS	ISPOS	
+1	TRA	*+2	IS A + SIGN
+2	TRA	YESPOS	
+3	CAS	ISNEG	
+4	TRA	*+2	- SING
+5	TRA	YESNEG	
+6	CAS	ISE	
+7	TRA	*+2	E
+8	TRA	YESE	
+9	CAS	ISDCPT	
+10	TRA	*+2	
+11	TRA	YESDPT	
+12	NZT	MASKER	
+13	TRA	ILEGAL	
+14	SUB	DELTA	
+15	ZET	1STNZ	
+16	TRA	STODIG	
+17	TRA	1STONE	
YESPOS	STZ	MANSNG	
+1	TRA	MORCLM	
YESNEG	STL	MANSNG	
+1	TRA	MORCLM	
YESDPT	SXA	DECCLM,2	
+1	STL	DPTIND	
+2	TRA	MORCLM	
1STONE	SXA	NZCOLM,2	
+1	STL	1STNZ	
+2	TRA	STODIG	
YESE	SXA	ECOLMN,2	E FOUND WILL TERMINATE SEARCH
+1	STL	EPART	
+2	LAC	ECOLMN,4	
	ICOLR	17,4	
+3	STL	43	
	ICHAR	ILEGAL,,8	
+6	STL	43	
	IINT	EEXP,,,3	
+10	STL	43	
GETOUT	CLA	DECCLM	
+1	CAS	NZCOLM	
+2	TRA	DECGR	DEC PT COLUMN IS LARGER
+3	HPR		OH REALLY
+4	CLA	NZCOLM	NON ZERO COLUMN IS LARGER
+5	SUB	DECCLM	
+6	SUB	ONE	
+7	SSM		
TESTE	NZT	EPART	WAS THERE AN EPART
+1	TRA	*+2	NO
+2	ADD	EEXP	YES ADD IN
+3	ADD	EXPONT	OCT 100

	+4	STO	EXPONT
	+5	TPL	ASSMBL
	+6	CLA	EXPØ
	+7	SUB	EXPCNT
	+8	STO	EXPCNT
	+9	TRA	ASSMBL
DECGR		CLA	DECCLM
	+1	SUB	NZCOLM
	+2	TRA	TESTE
ASSMBL		CLA	ZERC
	+1	ZET	MANSNG
	+2	CLA	ONE
	+3	ALS	7
	+4	CRA	EXPCNT
	+5	ALS	4
	+6	CRA	DIGIT
EXPDI		STO	**Ø
	+1	CLA	DIGIT+1
	+2	ALS	4
	+3	CRA	DIGIT+2
	+4	ALS	4
	+5	CRA	DIGIT+3
D234		STO	**Ø
	+1	CLA	DIGIT+4
	+2	ALS	4
	+3	CRA	DIGIT+5
	+4	ALS	4
	+5	CRA	DIGIT+6
D567		STO	**Ø
	+1	CLA	DIGIT+7
	+2	ALS	4
	+3	CRA	DIGIT+8
	+4	ALS	4
	+5	CRA	DIGIT+9
D89Ø		STO	**Ø
	+1	STZ	MASKER
		RETURN	SCANIT
	+2	TRA	SCANIT+1
		ICHAR	Ø
ILEGAL		STL	43
		OREADY	
	+4	STL	22
		OBCW	BADDEC,,1,2Ø
	+6	STL	22
	+1Ø	TSX	ERROR,4
	+11	STZ	MASKER
		RETURN	SCANIT
	+12	TRA	SCANIT+1
EVENUP		CAL	ICNTR
	+1	ANA	2BITS
	+2	SSM	
	+3	ADD	D4
	+4	ANA	2BITS
	+5	ADD	ICNTR

ASSEMBLE SIGN EXP + D1

NEG MANTISSA

	+6	STO	ICNTR
	+7	TRA	1,4
2BITS		PZE	3
D4		PZE	4
		BEGIN	1,7
RITOUT		TXL	*+5,0,0
REWB		REWB	8
	+1	SDLB	8
	+2	AXT	12,4
	+3	CLA	RIMAP+12,4
	+4	STO	SIMAP+12,4
	+5	TIX	*-2,4,1
	+6	AXT	1365,1
	+7	AXT	0,2
CLASIM		CLA	SIMAP,2
	+1	ALS	12
	+2	CRA	SIMAP+1,2
	+3	ALS	12
	+4	CRA	SIMAP+2,2
	+5	SLW	OREGIN+1365,1
	+6	TXI	*+1,2,-3
	+7	TIX	CLASIM,1,1
	+8	WTBB	8
	+9	RCHB	SCRIBE
	+10	TCOB	*
	+11	TRCB	REWB
	+12	REWB	8
		RETURN	RITOUT
	+13	TRA	RITOUT+1
ØTYPE		CAL	RITEKY
	+1	ARS	21
	+2	STO	FOROUT
	+3	ANA	M7777
	+4	STC	FCRMAP
	+5	TSX	OUTCOD,4
	+6	TSX	LCCMCH,4
	+7	TSX	MAPIT,4
	+8	TSX	WRLIST,4
	+9	CLA	ICNTR
	+10	ADD	ONE
	+11	STO	ICNTR
		ICHAR	0
BNKCRD		STL	43
		ICHAR	DILLY,,8
	+4	STL	43
	+8	NZT	IMODE
	+9	TRA	QDECM
		IOCTAL	TEMP,,16,4
	+10	STL	43
	+14	TRA	PICKUP
		IINT	TEMP,,16,4
QDECM		STL	43
PICKUP		CLA	TEMP
	+1	ANA	M7777

SUBROUTINE LINKAGE

EG BLS 0100 YYYY

+2	TRA	STOFCR	
	ICHAR	Ø	
DILLY	STL	43	
+4	TSX	FNDSYM,4	
+5	ANA	M7777	
+6	TRA	STOFCR	
1TYPE	CAL	RITEKY	EG PTA 0101 NO SYMBOLIC FIELD
+1	ARS	21	
STOFCR	STO	FCRCUT	
+1	ANA	M7777	
+2	STO	FCRMAP	
+3	TSX	OUTCOD,4	
+4	TSX	LCCMCH,4	
+5	TSX	MAPIT,4	
+6	TSX	WRLIST,4	
+7	TRA	REBACK	
	ICHAR	Ø	EG STP 015X
2TYPE	STL	43	
	ICHAR	SILLY,,8	
+4	STL	43	
	IINT	TEMP,,16,6	
+8	STL	43	
CLAM7	CLA	M7	
+1	ANS	TEMP	
CALRIT	CAL	RITEKY	
+1	ARS	21	
+2	CRA	TEMP	
+3	TRA	STOFCR	
	ICHAR	Ø	
SILLY	STL	43	
+4	TSX	FNDSYM,4	
+5	STO	TEMP	
+6	TRA	CLAM7	
	ICHAR	Ø	EG LDN 04XX
3TYPE	STL	43	
	ICHAR	NILLY,,8	
+4	STL	43	
+8	NZT	IMODE	
+9	TRA	QQDEC	
	IOCTAL	TEMP,,16,6	
+10	STL	43	
+14	TRA	CLAM77	
	IINT	TEMP,,16,6	
QQDEC	STL	43	
CLAM77	CLA	M77	
+1	CAS	TEMP	
+2	TRA	CALRIT	WITHIN RANGE
+3	TRA	CALRIT	
STZTEM	STZ	TEMP	
+1	TSX	OCRNG,4	
+2	TRA	CALRIT	
	ICHAR	Ø	
NILLY	STL	43	
+4	TSX	FNDSYM,4	

	+5	TNZ	JDD	
	+6	STZ	TEMP	
	+7	TRA	CALRIT	
JDD		STO	TEMP	
	+1	CAL	RITEKY	
	+2	ANA	TGMASK	
	+3	TZE	CLAM77	IS NOT FWD OR BACK
	+4	SUB	ITAG	
	+5	TNZ	ISBACK	REF BACKWARDS
	+6	CLA	TEMP	THIS REFS FWD
	+7	SUB	ICNTR	
TMIS		TMI	STZTEM	
	+1	STO	TEMP	
	+2	TRA	CLAM77	
ISBACK		CLA	ICNTR	
	+1	SUB	TEMP	
	+2	TRA	TMIS	
		ICHAR	Ø	
4TYPE		STL	43	
		ICHAR	TILLY,,8	
	+4	STL	43	
		IINT	TEMP,,16,1	
	+8	STL	43	
	+12	CLA	TEMP	
	+13	ALS	3	
	+14	STO	TEMP	
	+15	TRA	CALRIT	EG SLJ 77XØ
TILLY		TSX	FNDSYM,4	
	+1	TNZ	#+3	
	+2	STZ	TEMP	
	+3	TRA	CALRIT	
	+4	ALS	3	
	+5	STO	TEMP	
	+6	TRA	CLAM77	
5TYPE		TSX	FNDSYM,4	RELATIVE MODE
	+1	SUB	ICNTR	
	+2	TMI	ISREV	BACKWARD REFERENCE
	+3	STO	TEMP	
	+4	TRA	CLAM77	
ISREV		SSP		
	+1	STO	TEMP	
	+2	CAL	TGMASK	
	+3	ANA	RITEKY	
	+4	ALS	12	
	+5	ACL	RITEKY	
	+6	SLW	RITEKY	
	+7	TRA	CLAM77	
		BEGIN	1,4	
CUTCCD		TXL	#+3,Ø,Ø	SUBROUTINE LINKAGE
		OOCAL	FOROUT,,2Ø,4	
	+4	STL	22	
		RETURN	CUTCCD	
	+8	TRA	OUTCCD+1	
		BEGIN	1,4	

MAPIT TXL **3,0,0
 +4 CLA WHRMAP
 +5 ADD ICNTR
 +6 STA **2
 +7 CLA FORMAP
 +8 STO **0

SUBROUTINE LINKAGE

RETURN MAPIT
 +9 TRA MAPIT+1
 BEGIN 1,4

OCRNG TXL **3,0,0

SUBROUTINE LINKAGE

OREADY
 +4 STL 22
 OBCW RANGER,,1,20
 +6 STL 22
 +10 TSX ERROR,4
 RETURN CORNG
 +11 TRA OCRNG+1
 ICHAR GOOF,,8

OCTCRD STL 43
 IOCTAL TEMP,,16,4
 +4 STL 43
 +8 CLA TEMP
 +9 TRA STOFOR
 OREADY

GOOF STL 22
 OBCW GOON,,1,20
 +2 STL 22
 +6 TSX ERROR,4
 +7 CLA ZERO
 +8 TRA STOFOR

FORART AXT 3,1
 +1 TSX LCCMCH,4
 +2 TSX WRLIST,4

CLABAC CLA BACKTO+3,1
 +1 STO FORMAP
 +2 TSX MAPIT,4
 +3 CLA ICNTR
 +4 ADD ONE
 +5 STO ICNTR
 +6 TIX CLABAC,1,1
 +7 TRA CYCLE

INSTR EQU ICNTR

DETAIL

DETAIL

QSUBRA SSQR,BBB1,ENDSQR,DD1

FORSQR TSX EVENUP,4
 +1 CLA SSQR
 +2 STO SYMTAB,1
 +3 CLA ICNTR
 +4 STO LCCTAB,1
 +5 TXI **1,1,-1
 +6 SXA SYMCNT,1
 ICHAR BBB1,,8
 +7 STL 43

```

+8 TXL 44,0,12
+9 HTR BBB1,0
+10 PZE 8
    IINT TEMP,,1,6
+11 STL 43
+12 TXL 44,0,4
+13 STO TEMP,0
+14 PZE 1,,6
DD1 CLA ENDSQR
+1 ADD INSTR
+2 STO INSTR
+3 TRA READ2
    IBCW SYMTAB,1,1,1
BBB1 STL 43
+1 TXL 44,0,2
+2 HTR SYMTAB,1
+3 PZE 1,,1
    ICHAR ALLBNK,,2
+4 STL 43
+5 TXL 44,0,12
+6 HTR ALLBNK,0
+7 PZE 2
    ICHAR ISSYM,,8
+8 STL 43
+9 TXL 44,0,12
+10 HTR ISSYM,0
+11 PZE 8
+12 CLA ICNTR
+13 STO LCCTAB,1
+14 TXI *+1,1,-1
+15 SXA SYMCNT,1
+16 TRA DD1
    LIST
    QSUBRA SLOG,BBB2,ENDLOG,DD2
FORLOG TSX EVENUP,4
    QSUBRA SEXP,BBB3,ENDEXP,DD3
FOREXP TSX EVENUP,4
    QSUBRA SSIN,BBB4,ENDSIN,DD4
FORSIN TSX EVENUP,4
    QSUBRA SATN,BBB5,ENDATN,DD5
FORATN TSX EVENUP,4
    DETAIL
    QSUBRB ENDSQR
SQRFOR TSX EVENUP,4
+1 CLA WHRMAP
+2 ADD ICNTR
+3 ADD ENDSQR
+4 STA *+3
+5 LXA ENDSQR,4
+6 CLA ENDSQR,4
+7 STO **0,4
+8 TIX *-2,4,1
+9 TSX LOCMCH,4
+10 TSX LOCSIC,4

```

+11	TSX	WRLIST,4	
+12	CLA	ENDSQR	
+13	ADD	INSTR	
+14	STO	INSTR	
+15	TRA	CYCLE	
	LIST		
	QSUBRB	ENDLOG	
LOGFOR	TSX	EVENUP,4	
	QSUBRB	ENDEXP	
EXPFOR	TSX	EVENUP,4	
	QSUBRB	ENDSIN	
SINFOR	TSX	EVENUP,4	
	QSUBRB	ENDATN	
ATNFOR	TSX	EVENUP,4	
SSQR	BCI	1,SQR	
SLCG	BCI	1,LOG	
SEXP	BCI	1,EXP	
SSIN	BCI	1,SIN	
SATN	BCI	1,ATAN	
IMODE	PZE		
EQUASYM	PZE		
DOUBLE	BCI	9,	HAS BEEN MULTIPLY DEFINED....THE VALUE ASSIGNED
+9	BCI	9,	TO THE SYMBOL ON FIRST ENCOUNTER WILL BE USED...*****
+18	BCI	2,	*****
	LIST		
MODE	PZE		0= SICOM NON ZERO= MACHINE LANGUAGE
EXTMSK	BCI	1,MASK	
MASKER	PZE		NON ZERO FOR MASK CARD
DELTA	OCT	52	
SICOM	BCI	1,SICAP	
GOBACK	BCI	1,RETURN	RETURN TO MACHINE LANGUAGE
160A	BCI	1,MLAP	
THREE	DEC	3	
FOROUT	PZE		
FORMAP	PZE		
M7777	OCT	7777	
M7	OCT	7	
BADVAR	BCI	9,	THIS CARD HAS VIOLATED THE RULES FOR THE ADDITIVE SUBF
+9	BCI	9,	IELD.... IT WILL BE SET TO ZERO.. *****
+18	BCI	2,	*****
RANGER	BCI	9,	THE FOLLOWING CARD HAS AN OUT OF RANGE CONDITION. THE
+9	BCI	9,	XX OF THE EEXX FORMAT HAS BEEN REPLACED BY ZERO...*****
+18	BCI	2,	*****
M77	OCT	77	
GOCN	BCI	9,	THE VARIABLE FIELD OF THE FOLLOWING OCT CARD IS IN ERR
+9	BCI	9,	OR... ZERO HAS BEEN SUBSTITUTED.... *****
+18	BCI	2,	*****
ALLDUN	BCI	9,	THIS IS THE END OF THIS ASSEMBLY. REMOVE BINARY B8.
+9	BCI	9,	REMOVE LISTING TAPE A7 (UNLESS STACKING). PRESS START
+18	BCI	2,	FOR NEXT JOB
BACKTO	OCT	40	
+1	OCT	2006	
+2	OCT	30	
MINSTR	BCI	1,BLS	TYPE A TWO REGISTERS

+1	BCI	1,ATE
+2	BCI	1,ATX
+3	BCI	1,LDM
+4	BCI	1,LCC
+5	BCI	1,LCM
+6	BCI	1,LCC
+7	BCI	1,STM
+8	BCI	1,STC
+9	BCI	1,ADM
+10	BCI	1,ADC
+11	BCI	1,SBM
+12	BCI	1,SBC
+13	BCI	1,RAM
+14	BCI	1,RAC
+15	BCI	1,ACM
+16	BCI	1,ACC
+17	BCI	1,SRM
+18	BCI	1,SRC
+19	BCI	1,LPM
+20	BCI	1,LPC
+21	BCI	1,SCM
+22	BCI	1,SCC
+23	BCI	1,JPR
+24	BCI	1,IBI
+25	BCI	1,IBO
+26	BCI	1,EXC
BTYPE	BCI	1,PTA
+1	BCI	1,ETA
+2	BCI	1,CTA
+3	BCI	1,LDS
+4	BCI	1,LCS
+5	BCI	1,STS
+6	BCI	1,MUT
+7	BCI	1,MUH
+8	BCI	1,ADS
+9	BCI	1,SBS
+10	BCI	1,RAS
+11	BCI	1,AOS
+12	BCI	1,LS1
+13	BCI	1,LS2
+14	BCI	1,LS3
+15	BCI	1,LS6
+16	BCI	1,RS1
+17	BCI	1,RS2
+18	BCI	1,SRS
+19	BCI	1,LPS
+20	BCI	1,SCS
+21	BCI	1,CBC
+22	BCI	1,CIL
+23	BCI	1,INA
+24	BCI	1,OTA
+25	BCI	1,ERR
+26	BCI	1,HLT
+27	BCI	1,HPR

THIS TYPE (B) HAS NO VARIABLE FI

TYPE C FILL IN LO ORDER OCTAL DIGIT

+28	BCI	1,STP
+29	BCI	1,STE
+30	BCI	1,SRJ
+31	BCI	1,SIC
+32	BCI	1,IRJ
+33	BCI	1,SDC
+34	BCI	1,DRJ
+35	BCI	1,SID
+36	BCI	1,ACJ
+37	BCI	1,SBU
+38	BCI	1,NOP
+39	BCI	1,SLS
+40	BCI	1,LDN
+41	BCI	1,LDD
+42	BCI	1,LDI
+43	BCI	1,LDF
+44	BCI	1,LDB
+45	BCI	1,LCN
+46	BCI	1,LCD
+47	BCI	1,LCI
+48	BCI	1,LCF
+49	BCI	1,LCB
+50	BCI	1,STD
+51	BCI	1,STI
+52	BCI	1,STF
+53	BCI	1,STB
+54	BCI	1,HWI
+55	BCI	1,ADN
+56	BCI	1,ADD
+57	BCI	1,ADI
+58	BCI	1,ADF
+59	BCI	1,ADB
+60	BCI	1,SBN
+61	BCI	1,SBD
+62	BCI	1,SBI
+63	BCI	1,SBF
+64	BCI	1,SBB
+65	BCI	1,RAD
+66	BCI	1,RAI
+67	BCI	1,RAF
+68	BCI	1,RAB
+69	BCI	1,AOD
+70	BCI	1,ACI
+71	BCI	1,ACF
+72	BCI	1,AOB
+73	BCI	1,SRD
+74	BCI	1,SRI
+75	BCI	1,SRF
+76	BCI	1,SRB
+77	BCI	1,LPN
+78	BCI	1,LPD
+79	BCI	1,LPI
+80	BCI	1,LPF
+81	BCI	1,LPB

TYPE D FILL IN TWO LO ORDER OCTAL DIGI

+82	BCI	1,SCN
+83	BCI	1,SCD
+84	BCI	1,SCI
+85	BCI	1,SCF
+86	BCI	1,SCB
+87	BCI	1,ZJF
+88	BCI	1,NZF
+89	BCI	1,PJF
+90	BCI	1,NJF
+91	BCI	1,ZJB
+92	BCI	1,NZB
+93	BCI	1,PJB
+94	BCI	1,NJB
+95	BCI	1,JPI
+96	BCI	1,JFI
+97	BCI	1,INP
+98	BCI	1,CUT
+99	BCI	1,OTN
+100	BCI	1,EXF
+101	BCI	1,SJS
+102	BCI	1,SLJ
+103	BCI	1,ADR
+104	BCI	1,ACR
+105	BCI	1,LCR
+106	BCI	1,LDR
+107	BCI	1,LPR
+108	BCI	1,LSR
+109	BCI	1,NJR
+110	BCI	1,NZR
+111	BCI	1,PJR
+112	BCI	1,RAR
+113	BCI	1,SBR
+114	BCI	1,SCR
+115	BCI	1,SRR
+116	BCI	1,STR
+117	BCI	1,ZJR
+118	BCI	1,LSB
+119	BCI	1,LSD
+120	BCI	1,LSF
+121	BCI	1,LSI
+122	BCI	1,LSN
LSTINS	PZE	
	VFD	015/100,21/0
MCCODE		
	VFD	015/105,21/0
+1		
	VFD	015/106,21/0
+2		
	VFD	015/2100,21/0
+3		
	VFD	015/2200,21/0
+4		
	VFD	015/2500,21/0
+5		

	VFD	015/2600,21/0
+6	VFD	015/4100,21/0
+7	VFD	015/4200,21/0
+8	VFD	015/3100,21/0
+9	VFD	015/3200,21/0
+10	VFD	015/3500,21/0
+11	VFD	015/3600,21/0
+12	VFD	015/5100,21/0
+13	VFD	015/5200,21/0
+14	VFD	015/5500,21/0
+15	VFD	015/5600,21/0
+16	VFD	015/4500,21/0
+17	VFD	015/4600,21/0
+18	VFD	015/1100,21/0
+19	VFD	015/1200,21/0
+20	VFD	015/1500,21/0
+21	VFD	015/1600,21/0
+22	VFD	015/7100,21/0
+23	VFD	015/7200,21/0
+24	VFD	015/7300,21/0
+25	VFD	015/7500,21/0
+26	VFD	015/101,21/1
+27	VFD	015/107,21/1
+28	VFD	015/130,21/1
+29	VFD	015/2300,21/1
+30	VFD	015/2700,21/1
+31	VFD	015/4300,21/1
+32		

	VFD	015/112,21/1
+33	VFD	015/113,21/1
+34	VFD	015/3300,21/1
+35	VFD	015/3700,21/1
+36	VFD	015/5300,21/1
+37	VFD	015/5700,21/1
+38	VFD	015/102,21/1
+39	VFD	015/103,21/1
+40	VFD	015/110,21/1
+41	VFD	015/111,21/1
+42	VFD	015/114,21/1
+43	VFD	015/115,21/1
+44	VFD	015/4700,21/1
+45	VFD	015/1300,21/1
+46	VFD	015/1700,21/1
+47	VFD	015/104,21/1
+48	VFD	015/120,21/1
+49	VFD	015/7600,21/1
+50	VFD	015/7677,21/1
+51	VFD	015/,21/1
+52	VFD	015/7700,21/1
+53	VFD	015/7700,21/1
+54	VFD	015/150,21/2
+55	VFD	015/160,21/2
+56	VFD	015/10,21/2
+57	VFD	015/20,21/2
+58	VFD	015/30,21/2
+59		

	VFD	015/40,21/2
+60	VFD	015/50,21/2
+61	VFD	015/60,21/2
+62	VFD	015/70,21/2
+63	VFD	015/140,21/2
+64	VFD	015/,21/2
+65	VFD	015/7700,21/2
+66	VFD	015/400,21/3
+67	VFD	015/2000,21/3
+68	VFD	015/2100,21/3
+69	VFD	021/220001,15/3
+70	VFD	021/230002,15/3
+71	VFD	015/500,21/3
+72	VFD	015/2400,21/3
+73	VFD	015/2500,21/3
+74	VFD	021/260001,15/3
+75	VFD	021/270002,15/3
+76	VFD	015/4000,21/3
+77	VFD	015/4100,21/3
+78	VFD	021/420001,15/3
+79	VFD	021/430002,15/3
+80	VFD	015/7600,21/3
+81	VFD	015/600,21/3
+82	VFD	015/3000,21/3
+83	VFD	015/3100,21/3
+84	VFD	021/320001,15/3
+85	VFD	021/330002,15/3
+86		

	VFD	015/700,21/3
+87	VFD	015/3400,21/3
+88	VFD	015/3500,21/3
+89	VFD	021/360001,15/3
+90	VFD	021/370002,15/3
+91	VFD	015/5000,21/3
+92	VFD	015/5100,21/3
+93	VFD	021/520001,15/3
+94	VFD	021/530002,15/3
+95	VFD	015/5400,21/3
+96	VFD	015/5500,21/3
+97	VFD	021/560001,15/3
+98	VFD	021/570002,15/3
+99	VFD	015/4400,21/3
+100	VFD	015/4500,21/3
+101	VFD	021/460001,15/3
+102	VFD	021/470002,15/3
+103	VFD	015/200,21/3
+104	VFD	015/1000,21/3
+105	VFD	015/1100,21/3
+106	VFD	021/120001,15/3
+107	VFD	021/130002,15/3
+108	VFD	015/300,21/3
+109	VFD	015/1400,21/3
+110	VFD	015/1500,21/3
+111	VFD	021/160001,15/3
+112	VFD	021/170002,15/3
+113		

	VFD	021/600001,15/3
+114	VFD	021/610001,15/3
+115	VFD	021/620001,15/3
+116	VFD	021/630001,15/3
+117	VFD	021/640002,15/3
+118	VFD	021/650002,15/3
+119	VFD	021/660002,15/3
+120	VFD	021/670002,15/3
+121	VFD	015/7000,21/3
+122	VFD	021/710001,15/3
+123	VFD	021/720001,15/3
+124	VFD	021/730001,15/3
+125	VFD	015/7400,21/3
+126	VFD	015/7500,21/3
+127	VFD	015/7700,21/3
+128	VFD	015/7700,21/4
+129	VFD	021/320001,15/5
+130	VFD	021/560001,15/5
+131	VFD	021/260001,15/5
+132	VFD	021/220001,15/5
+133	VFD	021/120001,15/5
+134	VFD	021/160001,15/5
+135	VFD	021/630004,15/5
+136	VFD	021/610004,15/5
+137	VFD	021/620004,15/5
+138	VFD	021/520001,15/5
+139	VFD	021/360001,15/5
+140		

	VFD	021/160001,15/5
+141		
	VFD	021/460001,15/5
+142		
	VFD	021/420001,15/5
+143		
	VFD	021/610003,15/5
+144		
	VFD	021/170002,15/3
+145		
	VFD	021/140000,15/3
+146		
	VFD	021/160001,15/3
+147		
	VFD	021/150000,15/3
+148		
	VFD	021/30000,15/3
+149		
	VFD	015/1,21/1
LSTKEY		
SQRXXX	OCT	10,0,6,0,27,74,47,77
+8	OCT	52,30,53,23,54,30,45,77
+16	OCT	42,23,43,30,2,401,41,35
+24	OCT	5,75,2,64,7,75,4000,6025
+32	OCT	32,33,2,401,32,24,31,30
+40	OCT	22,74,37,25,7,0,11,0
+48	OCT	106,0,40,2024,6116,2023,6021,114
+56	OCT	740,4221,2023,201,4217,2200,100,4023
+64	OCT	2006,70,423,5002,2200,4374,70,423
+72	OCT	5002,6511,0,0,40,2303,5023,2304
+80	OCT	6520,401,5002,6523,0,0,4000,2004
+88	OCT	1060,1003,4450,2006,1506,4501,1106,2021
+96	OCT	3407,1562,1231,6022,2510,1464,43,2022
+104	OCT	3140,1167,542,2023
ENDSQR	PZE	*-SQRXXX
LOGXXX	OCT	10,0,6,0,74,74,37,30
+8	OCT	40,77,41,31,36,24,35,77
+16	OCT	4000,6025,37,77,40,25,41,30
+24	OCT	34,25,41,30,32,25,41,30
+32	OCT	30,25,41,30,26,25,41,30
+40	OCT	24,25,41,30,22,25,41,30
+48	OCT	20,25,41,30,12,25,5,31
+56	OCT	40,30,7,0,11,0,106,0
+64	OCT	1,0,0,2005,3140,1167,542,2003
+72	OCT	511,163,2606,6004,1440,2525,1444,2006
+80	OCT	2620,120,1501,2002,4123,2065,3131,2001
+88	OCT	4170,4471,1111,5761,1601,204,4022,1770
+96	OCT	3505,4520,2447,1767,4041,3422,3210,1771
+104	OCT	4631,3026,1100,2001,406,605,3467,2001
+112	OCT	2102,1226,4225,2002,3071,4211,3205,2010
+120	OCT	0,0,0,2021,60,2021,6003,2024
+128	OCT	6004,2200,4374,70,4312,4312,2023,3600
+136	OCT	100,4077,6034,6206,2200,6020,4323,2477
+144	OCT	6105,2200,2020,4330,2077,3200,3546,4077

```

+152 OCT 2177,4077,277,4076,2077,111,277,6013
+160 OCT 620,5346,2076,111,103,4353,2336,4023
+168 OCT 2006,70,2076,5360,6506,0
+174 OCT 0,0,0,0
ENDLOG PZE *-LOGXXX
EXPXXX OCT 10,0,6,0,66,77,100,701
+8 OCT 45,62,177,601,41,63,61,20
+16 OCT 62,27,55,77,56,20,53,31
+24 OCT 62,25,35,77,4000,6025,55,77
+32 OCT 16,401,12,601,52,23,6,601
+40 OCT 50,23,2,601,24,31,23,23
+48 OCT 4000,6030,1,601,4000,6025,4000,6025
+56 OCT 32,26,21,74,32,22,6,62
+64 OCT 1,101,25,24,7,0,11,0
+72 OCT 106,0,21,22,4,75,0,101
+80 OCT 6,75,3,74,42,74,0,0
+88 OCT 2200,4376,70,101,40,2023,6013,3704
+96 OCT 6005,2021,112,3020,6102,2021,103,103
+104 OCT 5207,2006,70,0,2124,30,442,2121
+112 OCT 0,0,400,2024,0,0,7400,2057
+120 OCT 0,0,0,2024,3462,2142,3526,2005
ENDEXP PZE *-EXPXXX
SINXXX OCT 132,1001,550,401,6,0,5,64
+8 OCT 6,0,101,33,6,0,101,25
+16 OCT 10,0,105,77,100,27,103,33
+24 OCT 2,62,1,601,42,60,4,501
+32 OCT 76,77,1,701,41,60,13,63
+40 OCT 1,1001,71,77,32,60,7,62
+48 OCT 1,601,34,60,4,62,1,601
+56 OCT 4000,6021,61,77,60,22,4000,6025
+64 OCT 60,77,27,25,30,30,55,25
+72 OCT 30,30,53,25,30,30,51,25
+80 OCT 30,30,47,25,30,30,45,25
+88 OCT 30,30,43,25,30,30,37,25
+96 OCT 7,0,11,0,106,0,1,101
+104 OCT 4,75,1,201,6,75,0,1760
+112 OCT 420,65,3210,5566,4442,562,3222,1625
+120 OCT 1065,4103,2630,5663,4110,2021,3004,1721
+128 OCT 465,3524,3201,5744,3044,1142,4551,1767
+136 OCT 4565,3100,2131,6006,1447,3626,2560,2021
+144 OCT 2061,2511,2621,2001,7760,7777,7777,6237
+152 OCT 0,0,0,1760,0,0,0,1760
+160 OCT 0,0,0,1760
ENDSIN PZE *-SINXXX
ATANXX OCT 10,0,6,0,43,74,1,301
+8 OCT 62,77,55,31,11,63,61,77
+16 OCT 56,22,51,25,1,601,55,23
+24 OCT 52,77,47,22,2,64,0,101
+32 OCT 72,77,45,22,4000,6025,45,77
+40 OCT 46,30,47,23,50,30,41,30
+48 OCT 50,23,51,30,36,30,51,23
+56 OCT 52,30,31,25,54,30,51,33
+64 OCT 20,74,7,0,11,0,106,0
+72 OCT 51,33,40,2024,4223,400,4024,2023

```

```

+80 OCT 3600,101,6307,2313,4326,2314,4327,2006
+88 OCT 70,5402,400,4334,400,4337,6607,0
+96 OCT 40,2302,4024,6614,3044,465,502,2004
+104 OCT 4027,4620,4446,2003,421,421,421,2001
+112 OCT 4401,2547,1064,1761,2600,3531,1470,2021
+120 OCT 1571,4071,2405,6001,2044,1220,3430,2022
+128 OCT 3511,1425,4450,6023,2470,1050,2006,2025
+136 OCT 140,4120,423,2023,2544,440,23,2002
+144 OCT 1447,3626,2560,2021,3174,52,3326,52
ENDATN PZE *-ATANXX
CREGIN BSS 1400,0
SCRIBE IOCD OREGIN,,1365
RETURN SCANIT
+1 TRA SCANIT+1
UNDEF BCI 9, IS AN UNDEFINED SYMBOL. IT WILL BE EQUATED TO
+9 BCI 3, ZERO.
+12 BCI 8, *****
NEWTEM BSS 5,H
NOCODE BCI 9, IS AN ILLEGAL OPCODE. A NOP HAS BEEN USED.
+9 BCI 3,
+12 BCI 8, *****
XXX BCI 1,XXXXXX
D19 DEC 19
RITEOP BCI 1,
RITEKY PZE 0
ACCODE BCI 1,$
AFIELD BCI 1,
DATA BSS 15,C
DIGIT BSS 10,C
MANSGN PZE
EXPONT PZE
EXP0 OCT 100
EXPSGN PZE
ISPOS BCI 1,00000+
ISNEG BCI 1,00000-
ISE BCI 1,00000E
ISBLNK BCI 1,00000
ISTNZ PZE
MANSET PZE ZERO= FIRST NON ZERO COLUMN NOT MET
EPART PZE ZERO = MANTISSA SIGN NOT SET
DECCLM PZE ZERO = NO E PART
NZCOLM PZE
TEN DEC 10
ECCLMN PZE
ISDCPT BCI 1,00000.
DPTIND PZE ZERO = NO DEC PT FOUND
D17 DEC 17
EEXP DEC 0 E PART OF EXPONENT
BADDEC BCI 9, THE FOLLOWING DEC CARD VIOLATES THE RULES FOR DATA IN
+9 BCI 9,PUT... A ZERO HAS BEEN STORED.
+18 BCI 2,
MCHLCC PZE 0
NAME BCI 9, THIS IS A 7090 TRANSFORMATION OF A SICOM PROGRAM WRIT
+9 BCI 9,TEN FOR THE 160-A COMPUTER BY SOMEONE WHO CHOOSES TO

```


+18	BCI	2, BE UNKNOWN.							
HEAD1	BSS	20,0							
HEAD2	BCI	9, MLOC	SILOC	K OP	ADDR		SYMBOL	OPCODE	VARIABLE FIE
+9	BCI	9, LD			COMMENTARY				PAGE
PAGE	BCI	2,							
NAMER	BCI	1, NAME							
PAGENO	PZE								
CASE1	CLA	22							
+1	SUB	ONE							
+2	STA	SWTCN+1							
	OHEAD	CASE1,,1,55,1							
+3	STL	22							
	OREADY								
+8	STL	22							
	OBCW	HEAD1,,1,20							
+10	STL	22							
	OSCRIB	CUTAPE,,20,1							
+14	STL	22							
	OREADY								
+18	STL	22							
+20	CLA	PAGENO							
+21	ADD	ONE							
+22	STO	PAGENO							
	OBCW	HEAD2,,1,20							
+23	STL	22							
	OINT	PAGENO,,109,2							
+27	STL	22							
	OSCRIB	CUTAPE,,20,1							
+31	STL	22							
	OREADY								
+35	STL	22							
	OBLANK	120,,1							
+37	STL	22							
	OSCRIB	CUTAPE,,20,1							
+40	STL	22							
SWTON	SWT	3							
+1	TRA	**0							
+2	WPDA								
+3	SPRA	1							
	OREADY								
+4	STL	22							
	OBCW	HEAD1,,1,20							
+6	STL	22							
	OSCRIB	PRINTR,,20							
+10	STL	22							
	OREADY								
+14	STL	22							
	OBCW	HEAD2,,1,20							
+16	STL	22							
	OINT	PAGENO,,109,2							
+20	STL	22							
	OSCRIB	PRINTR,,20							
+24	STL	22							
	OREADY								

+28	STL	22
	OBLANK	120,,1
+30	STL	22
	OSCRIB	PRINTR,,20
+33	STL	22
+37	TRA*	SWTON+1
SICLOC	DEC	0
TGMASK	PZE	,7
ITAG	PZE	,1
D1999	DEC	1999
D2000	DEC	2000
SIADDR	DEC	0
K	PZE	
C	PZE	
P	PZE	
A	PZE	
D1	PZE	
D2	PZE	
R	PZE	
ZERO	DEC	0
DECIMAL	DEC	1000,100,10
OCTGND	OCT	10000
BLANK	BCI	1,
ATE	DEC	12
OVRAGE	PZE	
8IND	OCT	4000
NINE	DEC	9
AD	DEC	
ACNUM	DEC	4096
ACLIST	BCI	1,\$000000
KOP	PZE	
RETURN	PZE	
ABLNK	BCI	1,000000
CARD	OCT	1321
PRINTR	OCT	1361
INUNIT	PZE	
OUTAPE	DEC	647
INTAPE	DEC	648
MEDTAP	DEC	1159
EQU	BCI	1,EQU
10THOU	DEC	10000
CNCFF	PZE	
INPUT	BSS	56,H
OUTPUT	BSS	56,H
WORK	BSS	14,H
TEMP	BSS	5,H
STAR	BCI	1,*000000
ORIGIN	BCI	1,ORG
DECDAT	BCI	1,DEC
END	BCI	1,END
CODE	PZE	
ICNTR	PZE	
TWO	PZE	2
DYNAMO	PZE	

NORMAL SICOM COMMAND = 7 CHARACTERS

CARD READER
PRINTER
CARD OR TAPE

B7 FOR COPY CARD INPUT

ZERO = ON LINE CARD INPUT

CNE	PZE	1	
SYMCNT	PZE		
	VFD	012/12,24/0	
LEADER			
ENABLE	OCT	2000	
CCTAL	BCI	1,OCT	
NUMBER	PZE		
SIMAP	BSS	4096,0	
WHRMAP	HTR	SIMAP	
WHRDAT	HTR	DATA	
REFADD	PZE	0,4	
RIMAP	DEC	18	READ IN PROGRAM
+1	OCT	0	
+2	DEC	21	READ TAE 0
+3	OCT	100	
+4	DEC	4	
+5	OCT	64	
G02	DEC	0	
+1	OCT	64	
+2	DEC	19	
+3	DEC	0	
+4	DEC	1	
+5	OCT	64	
BSS	BCI	1,BSS	
OPCODE	BCI	1,SIB	
+1	BCI	1,SID	
+2	BCI	1,SIL	
+3	BCI	1,DIB	
+4	BCI	1,IIB	
+5	BCI	1,SJB	
+6	BCI	1,SJD	
+7	BCI	1,SJL	
+8	BCI	1,DJB	
+9	BCI	1,IJB	
+10	BCI	1,CADM	
+11	BCI	1,CLS	
+12	BCI	1,CLA	
+13	BCI	1,IDVP	
+14	BCI	1,DVP	
+15	BCI	1,MPY	
+16	BCI	1,EXCH	
+17	BCI	1,EXTR	
+18	BCI	1,ADD	
+19	BCI	1,SUB	
+20	BCI	1,ADM	
+21	BCI	1,ISUB	
+22	BCI	1,RAD	
+23	BCI	1,CAS	
+24	BCI	1,OFLTB	
+25	BCI	1,OFLCR	
+26	BCI	1,OFXTB	
+27	BCI	1,OFXCR	
+28	BCI	1,OFORM	
+29	BCI	1,OCRTB	

+30	BCI	1,OTBNO
+31	BCI	1,OCMND
+32	BCI	1,WRFIL
+33	BCI	1,SRFIL
+34	BCI	1,GRPIN
+35	BCI	1,SNGIN
+36	BCI	1,IOCTAL
+37	BCI	1,OALPHA
+38	BCI	1,LDANR
+39	BCI	1,CMPANR
+40	BCI	1,MRGANR
+41	BCI	1,EXTANR
+42	BCI	1,TZE
+43	BCI	1,TNZ
+44	BCI	1,TPL
+45	BCI	1,TMI
+46	BCI	1,TSM4
+47	BCI	1,TSM5
+48	BCI	1,TSM6
+49	BCI	1,TNR7
+50	BCI	1,TSR
+51	BCI	1,TSJ1
+52	BCI	1,TSJ2
+53	BCI	1,TSJ3
+54	BCI	1,CALL
+55	BCI	1,TBR
+56	BCI	1,STOANR
+57	BCI	1,STO
+58	BCI	1,NCP
+59	BCI	1,STOP1
+60	BCI	1,STOP2
+61	BCI	1,HALT3
+62	BCI	1,HALT4
+63	BCI	1,HPRIN
+64	BCI	1,REL
+65	BCI	1,ABS
+66	BCI	1,FAST
+67	BCI	1,TRACE
+68	BCI	1,ZEROIR
+69	BCI	1,IFLEX
+70	BCI	1,OFLEX
+71	BCI	1,ITYPE
+72	BCI	1,OTYPE
+73	BCI	1,OPRINT
+74	BCI	1,SELBCD
+75	BCI	1,SELBIN
+76	BCI	1,BSR
+77	BCI	1,REW
+78	BCI	1,OTAPE
+79	BCI	1,ITAPE
+80	BCI	1,BLCPY
+81	BCI	1,BLC1R
+82	BCI	1,OOCTAL
+83	BCI	1,OSPEC

+84	BCI	1,ISPEC
+85	BCI	1,RZE
+86	BCI	1,RNZ
+87	BCI	1,RPL
+88	BCI	1,RMI
+89	BCI	1,RSM4
+90	BCI	1,RSM5
+91	BCI	1,RSM6
+92	BCI	1,RNR7
+93	BCI	1,RSR
+94	BCI	1,RSJ1
+95	BCI	1,RSJ2
+96	BCI	1,RSJ3
+97	BCI	1,OLOC
+98	BCI	1,PUSTOP
+99	BCI	1,CLAC
+100	BCI	1,CLSC
+101	BCI	1,IDVPC
+102	BCI	1,DVPC
+103	BCI	1,ADDC
+104	BCI	1,SUBC
+105	BCI	1,ISUBC
+106	BCI	1,SHIFT
+107	BCI	1,XEQ
+108	BCI	1,SETFWA
+109	BCI	1,SETLWA
+110	BCI	1,MPYC
+111	BCI	1,PXAIB
+112	BCI	1,PXAID
+113	BCI	1,PXAIL
+114	BCI	1,PXAJB
+115	BCI	1,PXAJD
+116	BCI	1,PXAJL
+117	BCI	1,PAXIB
+118	BCI	1,PAXID
+119	BCI	1,PAXIL
+120	BCI	1,PAXJB
+121	BCI	1,PAXJD
+122	BCI	1,PAXJL
ENDOP	PZE	
CPKEY	BCI	1,002000
SID	BCI	1,003000
SIL	BCI	1,004000
DIB	BCI	1,005000
IIB	BCI	1,006000
SJB	BCI	1,012000
SJD	BCI	1,013000
SJL	BCI	1,014000
DJB	BCI	1,015000
IJB	BCI	1,016000
CADM	BCI	1,020000
CLS	BCI	1,021000
CLA	BCI	1,022000
JVP	BCI	1,023000

THIS IS SIB 1ST OF TYPE 1 INSTRUCTIONS

DVP	BCI	1,024000
MPY	BCI	1,025000
EXCH	BCI	1,026000
EXTR	BCI	1,027000
ADD	BCI	1,030000
SUB	BCI	1,031000
ADM	BCI	1,032000
ISUB	BCI	1,033000
RAD	BCI	1,034000
CAS	BCI	1,035000
CFLTB	BCI	1,036000
CFLCR	BCI	1,037000
OFXTB	BCI	1,040000
OFXCR	BCI	1,041000
OFORM	BCI	1,042000
OCRTB	BCI	1,043000
OTBNO	BCI	1,044000
OCMND	BCI	1,045000
WFNMT	BCI	1,046000
SFNMT	BCI	1,047000
GRPIN	BCI	1,050000
SNGIN	BCI	1,051000
ROCTT	BCI	1,052000
ALNUO	BCI	1,053000
CLAANR	BCI	1,054000
CMPANR	BCI	1,055000
MRGANR	BCI	1,056000
EXTANR	BCI	1,057000
TZE	BCI	1,060000
TNZ	BCI	1,061000
TPL	BCI	1,062000
TMI	BCI	1,063000
TSM4	BCI	1,064000
TSM5	BCI	1,065000
TSM6	BCI	1,066000
TSM7	BCI	1,067000
TSR	BCI	1,070000
TSW1	BCI	1,071000
TSW2	BCI	1,072000
TSW3	BCI	1,073000
TML	BCI	1,074000
TBR	BCI	1,075000
STOANR	BCI	1,076000
STO	BCI	1,077000
NOP	BCI	1,000001
HLTSW1	BCI	1,001001
HLTSW2	BCI	1,002001
HALT3	BCI	1,003001
HALT4	BCI	1,004001
HLTGIN	BCI	1,005001
SELREL	BCI	1,006001
SELABS	BCI	1,007001
SELNTR	BCI	1,008001
RESTR	BCI	1,009001

END OF TYPE 0 INSTRUCTIONS
TYPE 1 INSTR OP CODE =DR

CLIR	BCI	1,010001
RDFLW	BCI	1,011001
WRFLW	BCI	1,012001
RDTPW	BCI	1,013001
WRTPW	BCI	1,014001
WRPRN	BCI	1,016001
SELTP	BCI	1,017001
SELMTD	BCI	1,018002
BSR	BCI	1,019004
REW	BCI	1,019001
WRMT2	BCI	1,020003
RDMT2	BCI	1,021003
BLCPY	BCI	1,024003
BLCLR	BCI	1,025003
POT	BCI	1,027003
PST	BCI	1,028003
RDST	BCI	1,029001
RZERC	BCI	1,060001
RNZERO	BCI	1,061001
RPCS	BCI	1,062001
RNEG	BCI	1,063001
RMK4	BCI	1,064001
RMK5	BCI	1,065001
RMK6	BCI	1,066001
RMK7	BCI	1,067001
RSR	BCI	1,070001
RSW1	BCI	1,071001
RSW2	BCI	1,072001
RSW3	BCI	1,073001
OLCC	BCI	1,075001
PLSTOP	BCI	1,079001
CLAC	BCI	1,101006
CLSC	BCI	1,201006
IDVPC	BCI	1,301006
DVPC	BCI	1,401006
ADDC	BCI	1,601006
SUBC	BCI	1,701006
ISUBC	BCI	1,801006
SHFTAC	BCI	1,901006
+1	BCI	1,X01006
SETTRC	BCI	1,Y01006
STPTRC	BCI	1,Z01006
MPYC	BCI	1,501006
PXIBA	BCI	1,022005
PXIDA	BCI	1,032005
PXILA	BCI	1,042005
PXJBA	BCI	1,122005
PXJDA	BCI	1,132005
PXJLA	BCI	1,142005
ATXIB	BCI	1,023005
ATXID	BCI	1,033005
ATXIL	BCI	1,043005
ATXJB	BCI	1,123005
ATXJD	BCI	1,133005

TYPE 2 INSTR

TYPE 6

ATXJL	BCI	1,143005
ENDKEY	PZE	1
SYMTAB	BSS	1000,H
LCCTAB	BSS	1000,C
	CRC	32256
	CORE	START,-1
DLMP	STL	2169
	CCRE	43,43
+4	STL	2169
	CCRE	22,22
+8	STL	2169
+12	TRA	110
	END	START

ENDCRD 0014	LOCSIC 0012	RITOUT 0023	1TYPE 0024
ENDEXP 0038	LOCTAB 0047	RNZERO *0046	2BITS 0023
ENDKEY 0047	LOGFOR 0028	SCANIT 0020	2TYPE 0024
ENDLCG 0038	LOGXXX 0037	SCRIBE 0039	3TYPE 0024
ENDSIN 0038	LOCKAT 0016	SELABS *0045	4TYPE 0025
ENDSQR 0037	LSTINS 0031	SELMTD *0046	5TYPE 0025
EQUCRD 0013	LSTKEY 0037	SELNTR *0045	8IND 0041
EQUSYM 0028	MANSET *0039	SELREL *0045	A 0041
EVENUP 0022	MANSGN 0039	SELTPP *0046	ABLNK 0041
EXPFOR 0028	MARKIT 0020	SETOFF 0001	ACNUM 0041
EXPONT 0039	MASKER 0028	SETTRC *0046	AD 0041
EXPSGN *0039	MCHLOC *0039	SHFTAC *0046	ADD *0045
EXPXXX 0038	MEDTAP 0041	SIADDR 0041	ADDC *0046
EXTANR *0045	MINSTR 0028	SICLOC *0041	ADDIT 0020
EXTMSK 0028	MORCLM 0021	SINFOR 0028	ADM *0045
FNDADD 0017	MOVEON 0018	SINXXX 0038	AFORK 0007
FNDBAK 0017	MRGANR *0045	SQRFOR 0027	AGN 0003
FNDCCM 0017	NEWTEM 0039	SQRXXX 0037	ALNUO *0045
FNDSYM 0015	NCCODE 0039	STABLE 0007	ALS3 0015
FORART 0026	NONOCT 0019	STOANR *0045	ASIF0 0008
FORATN 0027	NORMAL 0015	STOCMD 0015	ATE 0041
FOREXP 0027	NUMBER 0042	STODIG 0021	ATXIB *0046
FORLOG 0027	NXTONE 0011	STOFOR 0024	ATXID *0046
FORMAP 0028	NZCOLM 0039	STOKOP 0011	ATXIL *0046
FORMCH 0014	OBLANK 0013	STOREV 0015	ATXJB *0046
FOROUT 0028	OCTCRD 0026	STOROD 0015	ATXJD *0046
FORSIN 0027	OCTGND 0041	STPTRC *0046	ATXJL *0047
FORSQR 0026	OKTODR 0012	STUPID 0016	BBB1 0027
FOUNDX 0016	OPCODE 0042	STZTEM 0024	BBB2 0027
GETOUT 0021	OREDUN *0012	SYMBOL 0014	BBB3 0027
GOAHED 0008	OREGIN 0039	SYMCNT 0042	BBB4 0027
GOBACK 0028	ORGCRD 0013	SYMTAB 0047	BBB5 0027
GOLOCK 0007	ORIGIN 0041	TGMASK 0041	BLANK 0041
HLTGIN *0045	OTPEND 0012	UNPACK 0020	BLCLR *0046
HLTSW1 *0045	CUTAPE 0041	VARFRM 0016	BLCPY *0046
HLTSW2 *0045	OUTATR 0012	WHRDAT 0042	BSR *0046
ILEGAL 0022	OUTCOD 0025	WHRMAP 0042	BSS 0042
IMAGE1 0005	CUTPUT 0041	WRLIST 0012	BTYPE 0029
IMAGE2 *0005	OVRAGE 0041	YESADD 0018	CADM *0044
INDEXR 0020	PAGENO 0040	YESOPT 0021	CARD 0041
INTAPE 0041	PICKUP 0023	YESNEG 0021	CAS *0045
INUNIT 0041	PRINTR 0041	YESPOS 0021	CASE1 0040
IREDLN *0012	PRCTAG 0019	YESSYM 0008	CLA *0044
ISALPH 0019	PUSTOP *0046	YISOCT 0006	CLAC *0046
ISBACK 0025	RANGER 0028	YSCOMA 0019	CLAM7 0024
ISBLNK 0039	REBACK 0008	ZETADD *0017	CLAZ 0011
ISDCPT 0039	REFADD 0042	0TYPE 0023	CLIR *0046
LEADER *0042	RETURN 0041	160A 0028	CLS *0044
LITEIT 0003	RITEKY 0039	1STNZ 0039	CLSC *0046
LOCMCH 0012	RITEOP 0039	ITAG 0041	CODE 0041

HALT4 *0045
 HARD1 0019
 HEAD1 0040
 HEAD2 0040
 ICNTR 0041
 IDVP *0044
 IDVPC *0046
 IFDEC 0018
 IFOCT 0019
 IIB *0044
 IJB *0044
 IMODE 0028
 INPUT 0041
 INSTR 0026
 INTPL 0018
 ISAC 0016
 ISBSS 0003
 ISCOM 0020
 ISDEC 0004
 ISE 0039
 ISEND 0004
 ISNEG 0039
 ISOP 0003
 ISORG 0004
 ISPOS 0039
 ISREV 0025
 ISSYM 0003
 ISTAG 0019
 ISUB *0045
 ISUBC *0046
 JDD 0025
 K 0041
 KOP *0041
 LDQA 0010
 LOG1 0017
 LOG2 0017
 LOG3 0017
 LOOKC 0020
 LOOP 0010
 M7 0028
 M77 0028
 M7777 0028
 MAPIT 0026
 MCODE *0031
 MODE 0028
 MPY *0045
 MPYC *0046

MYOWN 0020
 NAME 0039
 NAMER 0040
 NILLY 0024
 NINE 0041
 NCEND *0012
 NCP *0045
 NOPE 0008
 NCT8 0011
 NCTAC 0015
 NCTK 0010
 NCWGO 0005
 NCWIN 0001
 O 0041
 OCMND *0045
 OCRTB *0045
 OCT12 0015
 OCTAL 0042
 OFLCR *0045
 OFLTB *0045
 OFORM *0045
 OFXCR *0045
 OFXTB *0045
 OLOC *0046
 ONE 0042
 CNOFF 0041
 OORNG 0026
 CPKEY *0044
 OTBNO *0045
 P 0041
 PAGE *0040
 PCT *0046
 PST *0046
 PXIBA *0046
 PXIDA *0046
 PXILA *0046
 PXJBA *0046
 PXJDA *0046
 PXJLA *0046
 QCECM 0023
 QCDEC 0024
 R 0041
 RAD *0045
 RAVEL 0017
 RDFLW *0046
 RDMT2 *0046
 RDST *0046

RDTPW *0046
 READ1 0001
 READ2 0001
 RESTR *0045
 REVRS 0013
 REW *0046
 REWB 0023
 RIMAP 0042
 RMK4 *0046
 RMK5 *0046
 RMK6 *0046
 RMK7 *0046
 RNEG *0046
 ROCTT *0045
 ROTAT 0018
 RPOS *0046
 RSR *0046
 RSW1 *0046
 RSW2 *0046
 RSW3 *0046
 RZERO *0046
 SATN 0028
 SEEIF 0011
 SEXP 0028
 SFNMT *0045
 SHFTR 0018
 SICOM 0028
 SID *0044
 SIL *0044
 SILLY 0024
 SIMAP 0042
 SIXX 0020
 SJB *0044
 SJD *0044
 SJL *0044
 SKIP *0002
 SLOG 0028
 SNGIN *0045
 SSIN 0028
 SSQR 0028
 STAR 0041
 START 0001
 STO *0045
 SUB *0045
 SUBC *0046
 SWTON 0040
 TBR *0045

TEMP 0041
 TEN 0039
 TESTE 0021
 THANX 0007
 THREE *0028
 TILLY 0029
 TMI *0045
 TMIS 0029
 TML *0045
 TNZ *0045
 TPL *0045
 TRYIT 0019
 TSM4 *0045
 TSM5 *0045
 TSM6 *0045
 TSM7 *0045
 TSR *0045
 TSW1 *0045
 TSW2 *0045
 TSW3 *0045
 TSXS 0008
 TWO 0041
 TYPE0 0007
 TYPE1 0008
 TYPE2 0009
 TYPE3 0009
 TYPE4 0009
 TYPE5 0010
 TYPE6 0010
 TZE *0045
 UNDEF 0039
 WFNMT *0045
 WIPE 0011
 WORK 0041
 WRFLW *0046
 WRMT2 *0046
 WRPRN *0046
 WRTPW *0046
 XXX *0039
 YESE 0021
 Z1 0006
 Z2 0006
 Z3 0006
 Z4 0006
 Z5 0006
 ZERO 0041

DISTRIBUTION LIST

Division 2

C. R. Wieser
S. H. Dodd

Group 21

W. Crowther
A. Curtiss (2)
J. Drinan
O. Fortier
H. Frachtman
D. Hafford
F. Heart
A. Mathiasen
R. Saliga
P. Smith
P. Stylos
R. Teoste
S. White

Group 22

M. Andrews
A. Arment
J. Avery
V. Brady
R. Carroll
N. Ciampa
D. Clapp
J. FitzGerald
A. Freed
R. Gauvreau (15)
A. Grometstein
J. Halberstein
C. Hardy
J. Hartogensis
P. Heffernan
R. Holland
C. Hopkins
D. Kramer
B. Lee
R. Leonard
D. Lovenvirth
A. MacGillivray
F. Mahoney
M. Malone
L. Meixsell
B. Nanni
J. Nolan
J. O'Brien
F. Oliver
D. Reiner
J. Rheinstein
K. Russell
B. Schafer
A. Smith
H. Sussman
J. Winett

Group 25

M. Check
E. Harvey
E. Hofmann
B. Moriarty
W. Vecchia
D. Wiggert

Group 26

H. Bostick
D. Underwood
H. Ziemann

Group 28

J. Arnov
D. Koniver
H. Meily
H. Peterson

Division 3

G. P. Dinneen

Group 32

B. Benn
H. Briscoe
R. Brown
J. Burdette
N. Donald
R. Martinson
J. Morriello
G. Price
K. Ralston
H. Shimmin

Group 33

L. Bird
V. Casaceli
E. Crowley
J. Crowley
R. Ingalls
L. Peterson
D. Quinn
G. Shannon
R. Shaputnic
M. Tausner
S. Wang
P. Willmann
F. Wilson
N. Reid

Group 34

J. Craig
P. Fleck
T. Goblick
S. McCarroll
R. Price
W. Smith

Group 35

B. Fowler
P. Miller
L. Tahmouh

Group 36

D. Hamilton
J. Max

Group 37

N. Holway

Group 314

R. Bröckelman
M. Meeks

Group 315

W. Danforth
M. Devane
L. Ricardi

Group 41

N. Doucett
J. Fielding

Group 42

J. Coglell
J. Dinsmore

Group 46

C. Freed
C. Muehe

Group 47

C. Berger
S. Catalano
H. Jones
R. Wishner

Group 48

C. Guay

Group 51

P. Fergus
J. Frankovich
J. Heart
S. Ornstein

Group 52

G. Blustein
S. Larkin
F. Nagy
M. Wendell

Group 55

F. Slagle
J. Slagle

Group 58

E. Freed
M. Goodman
L. Klem
C. McElwain
R. Mitchell
G. Mueser
J. Smith
M. Smith
A. Stowe
F. Trask
B. White
D. Yntema

Group 71

E. Boyce
W. Fanning
C. Pappas
D. Regillo
F. Wan

Group 75

D. Moore
P. Stetson

Group 81

M. Durgin
V. Mason
N. Rawson
S. Sillers
C. Work

Group 84

M. Dresselhaus
W. Mason

Group 22 Central Depository

J. FitzGerald

AFSSD

Col. L. S. Normar
Capt. J. Kieper (1)

Lockheed

H. Batten (5)

Philco WDL

F. Harris (5)

CDC

P. O. Cioffi (10)
A. Wilkonson (10)